UNIWERSYTET ŚLĄSKI
WYDZIAŁ INFORMATYKI I NAUKI O MATERIAŁACH
INSTYTUT INFORMATYKI

Rozprawa doktorska

mgr inż. Tomasz Orczyk

Klasyfikacja danych niekompletnych w oparciu o komitet klasyfikatorów

Promotor: dr hab. Piotr Porwik
Promotor pomocniczy: dr hab. inż. Bartłomiej Płaczek

Sosnowiec, 2018

Spis treści

Rozdział 1

Wstęp

Współczesne systemy informatyczne są w stanie gromadzić znaczne ilości danych. Dane te, po odpowiednim przetworzeniu, mogą być przydatne w budowie systemów rozpoznawania obiektów lub systemów wspomagania podejmowania decyzji.

Zadanie rozpoznawania polega na zaliczeniu obiektu do jednej z predefiniowanych klas. Obiekt posiada zawsze konkretną reprezentację i może mieć charakter rzeczywisty, ale może również posiadać sens abstrakcyjny. Niezależnie od tego podziału, rozpoznawany obiekt posiada atrybuty (cechy). Zamiast o obiektach możemy mówić o wektorach w przestrzeni cech, które są obrazami klasyfikowanych obiektów. Ze względów praktycznych, cechy obiektu są zazwyczaj uporządkowane i mogą być reprezentowane za pomocą wektora cech:

$$\mathbf{x} = [x^{(1)}, x^{(2)}, \ldots, x^{(F)}]. \tag{1.1}$$

Wektory $\mathbf{x}$ rozpinają przestrzeń cech X^F. Jest rzeczą oczywistą, że cechy $x^{(i)}$, $i = 1, \ldots, F$ mogą przybierać wartości z różnych przestrzeni. W świecie rzeczywistym mierzone sygnały mają przeważnie charakter ciągły, należą więc do przestrzeni funkcyjnych. Sygnał ciągły, modelowany różnymi funkcjami, można poddać dyskretyzacji. Tym sposobem funkcja zmiennej ciągłej może być reprezentowana ciągiem liczb, a ten ma swoją reprezentację w przestrzeni liniowej (wektorowej) [15]. Należy jednak pamiętać, że ciągowi danych dyskretnych odpowiadać może wiele sygnałów ciągłych. Przypadek skończonego zbioru wartości cech obiektu jest przypadkiem najczęściej występującym w praktyce. Dlatego w dalszej części pracy przyjęto, że $X^F \subseteq \mathbb{R}^F$.

Najważniejszą czynnością rozpoznawania obiektu jest jego klasyfikacja. Czynności rozpoznawania są przeważnie realizowane automatycznie za pomocą różnego rodzaju algorytmów. Klasyfikacja obiektów wymaga znajomości klas, do których należy przyporządkować poszczególne obiekty. Klasa obiektu jest pojęciem abstrakcyjnym. Na potrzeby niniejszego opisu przyjęto, że klasy obiektów są opisywane przez kolejne liczby naturalne. Identyfikator klasy skojarzony z zaobserwowanym wektorem cech obiektu określany będzie etykietą klasy obiektu lub w skrócie klasą obiektu. Niech zbiór identyfikatorów klas, który można utożsamiać z przestrze-

nią klas, oznaczony będzie następująco:

$$C = \{1, 2, \ldots, L\}, \tag{1.2}$$

gdzie L jest liczbą klas.
Oznaczmy algorytm klasyfikacji literą D. Dla powyższych założeń, algorytmy klasyfikacji D odwzorowują przestrzeń cech w przestrzeń klas:

$$D : X^F \to C. \tag{1.3}$$

Zależność (1.3) można również zapisać jako decyzję $D(\mathbf{x})$ klasyfikatora, na podstawie której obiekt reprezentowany wektorem cech $\mathbf{x}$ zostanie przypisany do klasy $c \in C$:

$$D(\mathbf{x}) = c, \quad \mathbf{x} \in X^F. \tag{1.4}$$

Klasyfikacja nadzorowana [38] polega na określeniu nieznanej klasy $c \in C$ wektora cech $\mathbf{x}$, z użyciem klasyfikatora utworzonego na podstawie zaobserwowanych w przeszłości par $(\mathbf{x}, c)$ o wartościach należących do $X^F \times C$.
Pary te nazywane są danymi uczącymi $(\mathbf{x}_1, c_1), (\mathbf{x}_2, c_2), \ldots, (\mathbf{x}_Q, c_Q)$, gdzie Q jest liczbą obserwacji w zbiorze uczącym, a $c_q \in C$ jest klasą skojarzoną z q-tym wektorem uczącym, który ma postać:

$$\mathbf{x}_q = [x_q^{(1)}, x_q^{(2)}, \ldots, x_q^{(F)}]. \tag{1.5}$$

Zbiór danych uczących będzie oznaczony jako $\mathbf{R}$:

$$\mathbf{R} = \{(\mathbf{x}_q, c_q)\}_{q=1}^{Q}. \tag{1.6}$$

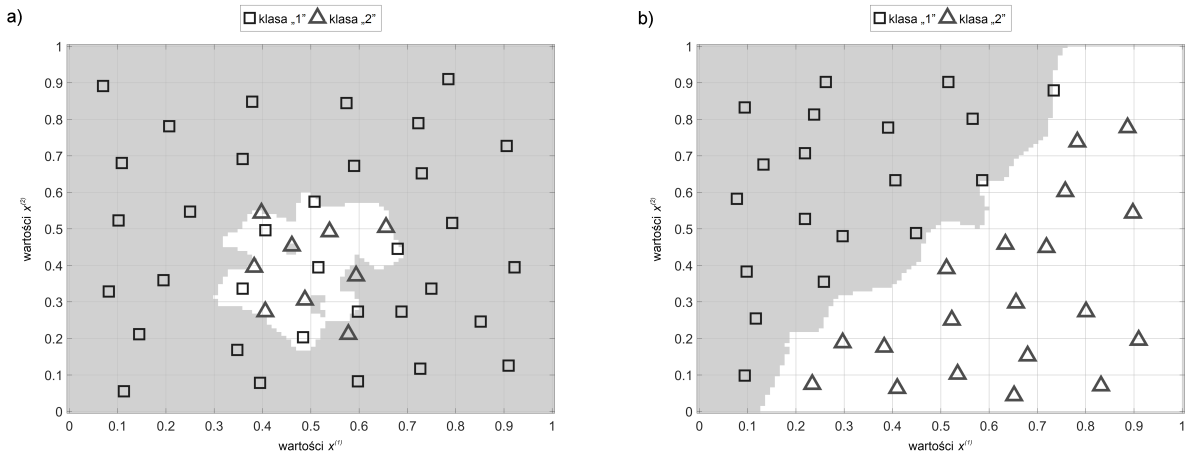
W celu zapewnia poprawność działania klasyfikatora, podczas tworzenia zbioru uczącego, należy zadbać o to aby został spełniony warunek, że każda klasa $c \in C$ jest reprezentowana w zbiorze danych uczących. Powyższy warunek można zapisać jako:

$$\forall_{c \in C} \exists_{(\mathbf{x}_q, c_q) \in \mathbf{R}} \left(c_q = c\right). \tag{1.7}$$

Istnieją również nienadzorowane techniki klasyfikacji, gdzie proces uczenia klasyfikatora jest zbędny, co oznacza, że klasyfikator musi samodzielnie ustalić związki występujące w danych i dokonać ich podziału na klasy. Zbiór danych składa się tutaj tylko z obiektów opisanych wektorem cech bez etykiet klas. Działanie takiego klasyfikatora polega na rozdzieleniu zbioru obiektów na dwa lub więcej podzbiory zawierające obiekty o podobnych cechach z wykorzystaniem określonej metryki lub miary podobieństwa. Zadaniem eksperta jest dokonanie interpretacji wydzielonych podzbiorów i przypisanie im odpowiednich znaczeń.

Należy zauważyć, że różnica pomiędzy klasyfikacją nadzorowaną i bez nadzoru sprowadza się do spostrzeżenia, że klasyfikator nadzorowany jest w stanie wyróżnić jedynie takie klasy, które zawarte były w zbiorze danych uczących.
W klasyfikacji nadzorowanej korzysta się z danych uczących, które są opisane przez eksperta dziedzinowego. Opis ten sprowadza się do przypisania każdego wektora cech do jednej z predefiniowanych kategorii, nazywanych klasami obiektu. Dane występują w postaci wektorów zawierających wyniki pomiaru (ilościowego lub jakościowego) szeregu cech. Wektor cech zawiera wyniki pomiarów zarejestrowane dla pojedynczego obiektu (np. pacjenta) i nazywany jest rekordem lub obiektem.

Zbiór uczący budowany jest na bazie historycznych danych, które mogą być niepełne (w poszczególnych rekordach niektóre z cech mogą być nieoznaczone). Zbiory uczące mogą również zawierać różną liczbę wektorów cech przypisanych do poszczególnych klas – mówimy wtedy o niezrównoważeniu klas w zbiorze uczącym. Niezrównoważenie klas utrudnia przygotowanie zbiorów uczących oraz komplikuje proces doboru i trenowania odpowiedniego klasyfikatora. Przykład niezrównoważonego zbioru danych zaprezentowano na Rys. 1.1a. W zbiorze tym występuje niezrównoważenie liczności obiektów należących do dwóch klas. Na rysunku tym zaznaczono również obszary klasyfikacji dla rozpoznawania klasy „□" – pole ciemne oraz klasy „△" – pole jasne. Łatwo zauważyć, że niektóre obiekty z klasy mniejszościowej „△" klasyfikowane są nieprawidłowo do większościowej klasy „□". Do klasyfikacji danych w omawianym przykładzie użyto klasyfikatora k-NN, ale problem ten występuje również w przypadku innych klasyfikatorów. W przypadku zrównoważonego zbioru danych, problem nieprawidłowej klasyfikacji występuje rzadziej, co zaprezentowano na Rys. 1.1b.



Rys. 1.1: Dane podlegające klasyfikacji: a) dane niezrównoważone, b) dane zrównoważone.

Na podstawie analizy danych zawartych w zbiorze uczącym można wyróżnić cztery rodzaje zbiorów danych. Zbiory zawierające:

a) Dane kompletne i równoliczne klasy przypadków.

b) Dane kompletne i niezrównoważone klasy przypadków.
c) Dane niekompletne i równoliczne klasy przypadków.
d) Dane niekompletne i niezrównoważone klasy przypadków.

Szczególnie trudnym do klasyfikacji rodzajem zbiorów danych są zbiory wymienione w punkcie d) powyższego zestawienia. Typowym przedstawicielem takiej kolekcji danych są zbiory medyczne. Niekompletność cech może wynikać na przykład ze sposobu pracy laboratorium analitycznego, gdzie niektóre parametry krwi obwodowej nie są oznaczane, są oznaczane różnymi metodami lub według różnych norm. Niezrównoważenie klasy wynika z częstotliwości występowania analizowanego schorzenia. W niniejszej pracy takie zbiory będą także analizowane.

Istnieje szereg metod pozwalających na uzupełnianie brakujących danych w zbiorach uczących [22, 28, 29, 55, 56]. Niekiedy nie pozostają one bez wpływu na wiarygodność klasyfikatora uczonego na tak zmodyfikowanych danych [44]. Skrajnym przykładem nieprawidłowego uzupełniania brakujących danych medycznych może być użycie metody Expectation Maximization (EM)[54] do uzupełniania danych rzeczywistej kolekcji pomiarów medycznych związanych z wykrywaniem włóknienia wątroby na podstawie analizy krwi obwodowej pacjenta. Zastosowanie metody EM powoduje, że brakujące dane są uzupełnianie wartościami z poza dopuszczalnego zakresu zmienności mierzonego parametru. W Tabeli 1.1 przedstawiono fragment wyżej wspomnianej bazy danych z uzupełnionymi brakującymi wartościami metodą EM w programie WEKA [24]. Pogrubioną czcionką zaznaczono miejsca nieprawidłowego uzupełnienia danych. Jedno z takich miejsc wskazuje pole z ujemną wartością oznaczającą liczbę białych krwinek (neutrofili), co z punktu widzenia diagnostyki medycznej jest dyskwalifikujące. Przykładem nieprawidłowego uzupełniania danych w tej samej bazie jest również metoda regresji liniowej z modułu *mice* programu *R* (Tabela 1.2). W tym przypadku, wartości ujemne pojawiały się m.in. w kolumnie opisującej wagę pacjenta. Inne problemy związane z uzupełnianiem danych (np. korelacja uzupełnianych wartości z klasą obiektu), zarówno treningowych jak i klasyfikowanych, zostały opisane w pracy [44].

Tabela 1.1: Przykład nieprawidłowego uzupełniania danych metodą EM (WEKA).

ALAT	BIL	ALP	CRP	TG	K	GGTP	KREA	Na	GLU	KwMocz	TP	Albu	CHOL	Neutr
76	0,6	89	**-1,89**	116	4,2	95	0,8	139	106	7,1	6,8	4,4	131	2,10
52	0,7	86	0,56	82	4,2	31	0,9	138	71	6,3	8,1	4,4	200	3,29
53	0,7	83	1,95	172	4,5	79	1,0	140	84	5,0	7,3	4,4	198	4,58
64	1,2	31	1,20	131	4,5	14	0,9	138	91	5,3	7,4	4,9	184	3,45
158	1,4	63	0,41	183	4,0	114	1,2	138	110	7,4	7,7	4,1	174	1,86
44	1,2	61	1,55	55	3,9	59	1,0	138	97	7,7	9,2	4,8	187	4,16
55	0,6	88	0,50	95	4,2	361	1,0	137	87	5,8	8,7	4,7	217	5,98
222	1,2	108	2,10	192	4,5	147	0,9	137	123	7,3	6,5	4,6	179	1,99
168	0,9	131	2,03	65	4,2	225	0,9	140	95	5,8	7,8	4,5	208	2,64
117	3,0	128	3,73	**-47**	3,9	139	1,0	140	91	4,7	7,3	2,7	141	1,08
59	0,9	76	0,36	413	4,0	123	1,0	136	125	7,1	6,4	4,6	159	3,05
64	1,1	87	0,95	113	4,2	130	0,9	137	90	6,2	8,0	4,0	204	3,61
76	0,6	94	0,48	154	4,6	171	1,0	136	86	7,2	7,7	4,0	194	2,63
87	0,9	69	0,22	88	3,6	135	1,3	140	98	5,6	8,7	4,1	193	5,77
77	1,8	93	2,60	200	4,2	115	1,2	136	107	4,1	8,6	4,5	165	4,37
70	1,4	64	0,40	150	4,5	59	1,3	141	94	6,5	7,4	5,0	198	2,57
45	1,3	53	1,40	179	4,5	137	1,1	143	87	6,3	7,2	4,7	157	4,31
37	0,7	110	0,66	62	3,8	146	0,9	138	91	5,6	8,4	4,6	198	5,59
24	6,0	103	6,77	259	4,5	28	1,2	129	51	6,1	7,1	4,9	132	1,94
19	2,8	160	4,79	59	4,6	**-14**	0,7	134	73	2,7	6,2	2,1	148	**-0,45**
90	2,1	141	4,21	80	4,1	35	0,9	132	106	3,5	6,6	2,9	152	0,24
67	2,0	177	1,34	96	4,5	425	0,6	138	80	5,1	7,1	4,4	156	2,35
117	1,6	95	1,11	126	4,2	124	1,1	138	95	5,6	8,0	4,5	176	4,34
72	3,2	83	6,94	193	4,5	124	0,7	138	83	3,4	6,3	3,7	197	0,86
13	1,9	219	8,19	375	6,2	161	1,0	126	72	3,6	5,2	2,7	171	0,72
293	1,0	65	0,16	194	4,3	127	1,1	138	129	6,2	7,4	4,6	195	2,33
42	1,8	108	2,58	176	4,2	126	1,0	133	139	6,3	7,2	4,1	191	2,05
48	1,2	74	1,16	118	4,2	40	0,8	142	104	5,1	6,5	4,2	165	1,69
182	3,7	123	2,46	**-134**	4,0	76	0,7	137	70	12,6	7,5	3,8	98	3,36
19	2,3	151	3,05	43	4,9	32	0,6	139	97	4,9	6,2	3,2	153	0,22
27	1,6	57	2,49	90	4,7	47	0,7	139	74	6,0	7,2	4,6	165	3,13
40	0,8	109	0,65	89	4,5	40	1,0	143	109	5,9	6,8	4,4	116	4,95
73	2,5	211	2,39	86	4,4	30	0,1	135	99	2,5	6,6	3,6	164	**-0,94**
245	1,0	77	**-0,22**	**-9**	3,9	133	0,6	138	71	5,3	8,2	4,1	138	0,50

Tabela 1.2: Przykład nieprawidłowego uzupełniania danych metodą regresji liniowej (R).

Waga	HB	RBC	PLT	WBC	PTP	APTT	INR	PT	ASPT	ALAT	BIL	ALP	CRP	TG
116	15,2	5	162,00	6	93,0	35	1,1	11	24	26,0	0,8	87,0	**-1**	85,90
121	17,7	6	231,00	9	126,0	25	0,9	10	58	115,0	1,0	99,0	0	110,30
133	16,5	6	165,00	7	104,0	31	1,0	10	26	33,0	1,2	68,0	**-1**	128,90
56	14,8	5	185,00	6	111,8	31	1,0	10	26	31,0	0,6	96,0	1	128,80
43	15,4	5	255,00	8	110,2	36	1,1	11	42	79,0	0,4	92,0	0	93,80
47	16,0	5	156,00	6	84,9	34	1,2	13	184	240,0	0,7	92,0	2	82,00
45	13,6	5	162,00	6	109,4	29	1,0	10	56	58,0	0,8	139,0	0	76,00
91	16,9	5	158,00	7	105,5	36	1,0	10	41	44,0	1,5	44,0	0	127,00
66	12,4	5	207,00	6	117,0	26	1,0	11	23	23,0	0,9	60,0	1	136,00
90	14,3	5	187,00	6	85,0	33	1,1	13	46	39,0	0,9	85,0	0	91,00
97	15,4	5	181,00	5	106,0	33	1,0	11	25	52,0	1,0	49,0	0	138,40
60	14,0	5	250,00	5	130,0	27	0,9	12	47	70,0	1,3	76,0	5	117,80
21	14,2	5	270,00	7	126,9	29	1,0	10	39	56,0	1,0	76,0	3	142,00
40	15,3	5	229,00	5	106,9	30	1,0	11	21	27,0	0,8	51,0	1	192,80
58	13,6	5	214,00	5	93,0	30	1,1	11	24	32,0	0,9	60,0	0	116,40
103	15,5	5	168,00	5	98,8	35	1,1	11	29	46,0	0,9	66,0	**-1**	123,00
114	15,6	5	232,00	9	93,0	33	1,1	13	70	29,0	1,5	60,0	2	69,70
91	13,3	4	186,00	7	77,8	34	1,3	14	30	28,0	0,4	78,0	-2	96,20
-16	12,3	4	235,00	6	112,0	35	1,0	11	52	66,0	0,6	65,0	3	232,00
-41	13,7	5	210,00	5	128,0	26	0,9	10	32	41,0	0,8	108,0	10	291,90
70	12,8	4	227,00	5	118,0	29	1,0	11	38	48,0	0,7	89,0	1	69,00
93	14,8	5	167,00	8	105,0	30	1,0	11	93	126,0	1,8	112,0	5	133,20
50	14,8	5	186,00	6	135,0	27	0,9	10	31	44,0	2,1	55,0	2	143,40
55	14,9	5	207,00	6	94,7	36	1,2	12	34	45,0	0,9	93,0	0	89,90
34	12,8	5	238,00	5	115,0	27	0,9	11	37	43,0	0,5	42,0	3	197,40
91	14,1	5	214,00	5	91,0	37	1,1	13	34	32,0	1,3	83,0	1	135,40
18	14,7	5	281,00	3	106,0	34	1,1	11	34	36,0	1,4	66,0	2	107,20
-27	14,4	5	335,00	5	103,0	27	1,0	11	186	334,0	0,6	99,0	4	176,90

Klasyfikator najlepiej uczyć, a później testować na rzeczywistym zbiorze danych, jednak nie zawsze jest to możliwe. Aby odzwierciedlić problemy występujące w klasyfikacji rzeczywistych zbiorów danych, można posłużyć się ogólnodostępnymi zbiorami danych odniesienia (np. w *Machine Learning Repository*). Są to tzw. zbiory testowe (benchmarkowe), które zawierają oprócz danych wygenerowanych sztucznie także dane pochodzące z procesów rzeczywistych. W oparciu o zbiory benchmarkowe można wygenerować zbiory zdegradowane, poprzez usunięcie losowych instancji aby zaburzyć równowagę liczności klas oraz usunięcie wartości losowo wybranych cech z losowo wybranych instancji w celu wprowadzenia losowo brakujących wartości.

Zaproponowany w niniejszej pracy klasyfikator umożliwia uzyskanie wysokiej dokładności klasyfikacji danych niepełnych oraz niezrównoważonych. Proponowana metoda może być wykorzystywana nawet wtedy, gdy wymienione powyżej niekorzystne przypadki rozkładu danych występują równocześnie. Rozwiązanie będzie zaprezentowane od strony teoretycznej jak i eksperymentalnej.

1.1 Teza pracy

Możliwe jest utrzymanie dokładności klasyfikacji na danych niepełnych poprzez wyłonienie komitetu klasyfikatorów działających w oparciu o wstępnie wyselekcjonowane cechy.

Celem proponowanego rozwiązania ma być utrzymanie dokładności klasyfikacji w miarę rozbudowy zbioru danych uczących i zmiany proporcji częstości występowania przypadków należących do poszczególnych klas oraz pojawiania się brakujących wartości w wektorach danych referencyjnych jak i w danych do klasyfikacji. Z uwagi na przyjęte założenie o ciągłej rozbudowie repozytorium (przypadków referencyjnych) w pracy skupiono się nad klasyfikatorami leniwymi, tj. takimi, które nie tworzą modelu w oparciu o dane uczące lecz szukają rozwiązania wśród danych referencyjnych dopiero w momencie pojawienia się wektora danych do sklasyfikowania. Za takim wyborem przemawia też fakt, iż komitety klasyfikatorów są bardziej efektywne kiedy tworzą je słabe klasyfikatory (tj. takie o skuteczności tylko nieco lepszej od klasyfikatora losowego) [71].

1.2 Cele pracy

Zdefiniowano następujące cele pracy:

1. **Oszacowanie wpływu brakujących lub usuniętych cech obiektu na jakość klasyfikacji.** W przypadku wykorzystywania danych historycznych jako zbioru uczącego mogą w nim występować brakujące wartości, których nie da się uzupełnić. Odrzucanie niekompletnych wektorów danych wiąże się z utratą informacji. Usunięcie brakującej cechy w pozostałych wektorach danych również mogłoby spowodować usunięcie potencjalnie istotnych danych. Istotne będzie więc określenie wpływu liczby brakujących wartości na jakość klasyfikacji.

2. **Opracowanie struktury komitetu klasyfikatorów.** Z uwagi na strukturę komitetu można je podzielić na równoległe, szeregowe (kaskadowe) oraz iteracyjne. Możliwe są też rozwiązania mieszane. Konieczne jest więc dokonanie wyboru struktury komitetu adekwatnej do charakterystyki klasyfikowanych danych.

3. **Wybór klasyfikatorów działających w komitecie.** Komitety buduje się zazwyczaj w oparciu o słabe klasyfikatory. Wśród nich można wyróżnić komitety klasyfikatorów opartych o ten sam algorytm klasyfikacji, ale operujących na różnych podzbiorach cech, oraz na komitety klasyfikatorów opartych o różne algorytmy klasyfikacji. Rzeczą istotną

jest zatem wybór klasyfikatora lub klasyfikatorów, które zostaną użyte do budowy komitetu.

4. **Opracowanie algorytmu podejmowania decyzji (fusera) komitetu klasyfikatorów.** Ważnym elementem, zwłaszcza w przypadku komitetów równoległych, jest algorytm wyłaniający ostateczną decyzję w oparciu o dane zwrócone przez poszczególne klasyfikatory wchodzące w skład komitetu.

5. **Wybór cech dystynktywnych dla poszczególnych klas obiektów.** Usunięcie pewnych cech, które nie mają wpływu na wynik klasyfikacji, może nie tylko ograniczyć złożoność algorytmu klasyfikującego, a co za tym idzie zredukować czas klasyfikacji i zajętość pamięci, ale również podnieść dokładność klasyfikacji. Alternatywnie możliwe jest określenie przydatności poszczególnych cech do rozróżniania klas i użycie tej wiedzy do ważenia decyzji klasyfikatorów w zależności od cech, na których działały i wyniku, jaki zwróciły.

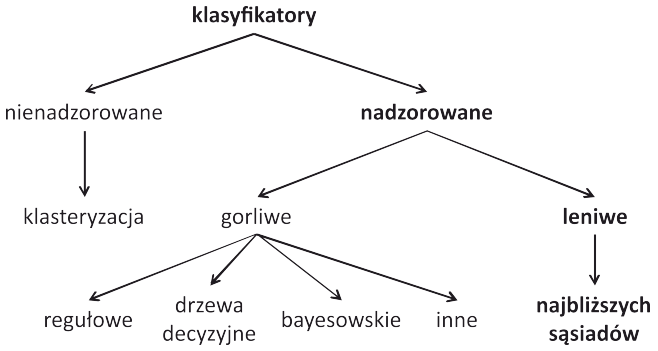
6. **Testowanie opracowanego systemu na danych rzeczywistych.** Każdy system wspomagania decyzji projektowany jest z myślą o konkretnym zastosowaniu. Aby potwierdzić przydatność zaprezentowanego rozwiązania konieczne jest jego przetestowanie na rzeczywistych danych, z jakimi docelowo będzie miał pracować.

7. **Weryfikacja przydatności opracowanego klasyfikatora do budowy systemu oceny stopnia włóknienia wątroby u pacjentów z wirusowym zapaleniem wątroby typu C w oparciu o analizę parametrów krwi obwodowej.** Ostatnim celem pracy będzie sprawdzenie przydatności opracowanego klasyfikatora jako elementu oprogramowania wspierającego pracę lekarzy hepatologów w ocenie stopnia zaawansowania choroby wywoływanej przez wirusa zapalenia wątroby. Celem tego oprogramowania będzie ograniczenie potrzeby wykonywania biopsji wątroby, która jest zabiegiem inwazyjnym, wiążącym się potrzebą hospitalizacji pacjenta i niosącym ryzyko poważnych powikłań zdrowotnych.

Rozdział 2

Przegląd wybranych metod klasyfikacji

Klasyfikator można utożsamiać z algorytmem, który na podstawie analizy cech obiektu przypisuje go do odpowiedniej klasy. Wśród klasyfikatorów można wyróżnić szereg grup (Rys. 2.1).



Rys. 2.1: Systematyka algorytmów klasyfikacji.

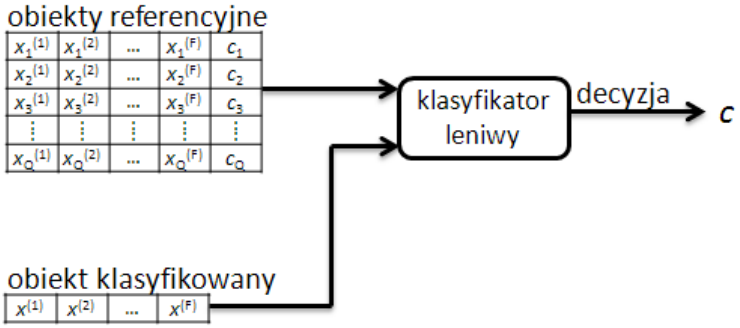
W niniejszej pracy rozpatrywane są modyfikacje algorytmów leniwej klasyfikacji nadzorowanej opartej na metodzie najbliższych sąsiadów. Uzasadnienie tego wyboru przedstawione będzie w Rozdziale 7. Obszar zagadnień, których dotyczy praca został wyróżniony na Rys. 2.1 pogrubioną czcionką.

2.1 Klasyfikatory leniwe

Klasyfikatory leniwe nie wymagają wstępnego tworzenia modelu matematycznego na podstawie dostępnych danych, gdyż klasyfikacja odbywa się bezpośrednio w oparciu o zgromadzone dane uczące (Rys. 2.2). Z uwagi na brak etapu uczenia, w odniesieniu do klasyfikatorów leniwych, zamiennie z pojęciem zbioru uczącego, stosuje się termin zbiór referencyjny. Zaletami klasyfikatorów leniwych są m.in. natychmiastowa gotowość algorytmu do przeprowadzenia klasyfikacji oraz możliwość uzupełniania zbioru referencyjnego w dowolnym momencie pracy. Wadą jest relatywnie długi czas klasyfikacji.

Na Rysunku 2.2 przedstawiono przykład przepływu danych w klasyfikatorze leniwym. f-tą cechę q-tego obiektu oznaczono jako $x_q^{(f)} \in X^F$, klasę tego obiektu oznaczono jako $c_q \in C$.

W przedstawionym na rysunku procesie klasyfikacji zbiór referencyjny stanowi Q obiektów opisanych przez F cech, z których każdy należy do jednej z klas zbioru C.



Rys. 2.2: Przepływ danych w klasyfikatorze leniwym.

Metoda k-najbliższych sąsiadów (k-NN)

Metoda k najbliższych sąsiadów (*ang.* k-Nearest Neighbors) [31] jest przykładem klasyfikatora leniwego. W tej strategii obiekt poddawany procesowi klasyfikacji zaliczany jest do klasy, do której należy większość spośród jego k najbliższych sąsiadów. Jako najbliższych sąsiadów klasyfikowanego obiektu należy rozumieć najbardziej podobne do niego, pod względem przyjętej miary podobieństwa, obiekty ze zbioru referencyjnego. Nie jest zatem konieczne tworzenie uogólnionego modelu danych, ani poszukiwanie sposobu rzutowania przestrzeni cech pozwalającego na stworzenie funkcji opisującej granice pomiędzy obiektami należącymi do różnych klas dla całego zbioru danych uczących. Klasyfikacja odbywa się wprost, w oparciu o głosowanie klas obiektów najbardziej podobnych do obiektu klasyfikowanego. Metoda k-NN definiuje estymator prawdopodobieństwa *a posteriori* P przynależności wektora cech obiektu $\mathbf{x}$ do klasy c. Oznaczmy wielowymiarowy zbiór wektorów referencyjnych, do których ma dostęp klasyfikator przez $A = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_Q\}$.
Wówczas estymator prawdopodobieństwa *a posteriori* przynależności wektora $\mathbf{x}$ do klasy c wyznacza się na podstawie zbioru k najbliższych wektorów referencyjnych, zgodnie ze wzorem:

$$P(c|\mathbf{x}) = \frac{1}{k} \sum_{q=1}^{Q} I\left(\rho(\mathbf{x},\mathbf{x}_q) \leq \rho(\mathbf{x},\mathbf{x}_k)\right) I\left(c_q = c\right), \tag{2.1}$$

gdzie $\mathbf{x}_k \in A$ jest k-tym co do odległości od $\mathbf{x}$ wektorem referencyjnym, ρ jest przyjętą miarą odległości, natomiast I jest funkcją indykatorową, definiowaną następująco:

$$I(\omega) = \begin{cases} 1 & \text{gdy warunek } \omega \text{ jest prawdziwy} \\ 0 & \text{w przeciwnym przypadku} \end{cases}. \tag{2.2}$$

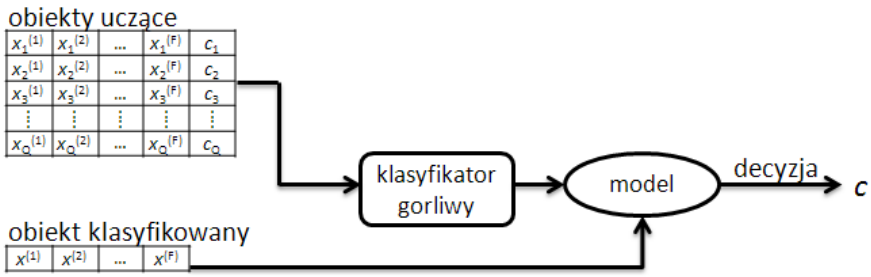
Ostatecznie klasyfikator podejmuje decyzję o przynależności obiektu reprezentowanego wektorem cech $\mathbf{x}$ do klasy c na podstawie formuły

$$D_{kNN}(\mathbf{x}) = \underset{c \in C}{\operatorname{argmax}}(P(c|\mathbf{x})). \tag{2.3}$$

Wybór używanej w (2.1) miary odległości jest arbitralny i zależy przede wszystkim od charakteru danych. Wybrane miary odległości, wraz z bardziej szczegółowym opisem metody k-NN i jej modyfikacji zawarte zostały w Rozdziale 6.

2.2 Klasyfikatory gorliwe

Klasyfikatory gorliwe są to klasyfikatory budujące modele opisujące zależności pomiędzy klasą, a cechami obiektów. Po stworzeniu modelu, klasyfikatory tego typu nie korzystają już z danych uczących i nie muszą ich przechowywać, chyba że proces uczenia ma być ponawiany w przyszłości z uwzględnieniem dodatkowych, nowo poznanych wektorów danych. Klasyfikacja, w przypadku klasyfikatorów gorliwych, odbywa się w oparciu o wcześniej wygenerowany model (hipotezę globalną), co wymaga wygenerowania uogólnionych reguł obejmujących całą przestrzeń cech. Zalety tego typu klasyfikatorów to krótki czas klasyfikacji i reprezentacja danych przy pomocy modelu, wadą jest długi czas uczenia się.



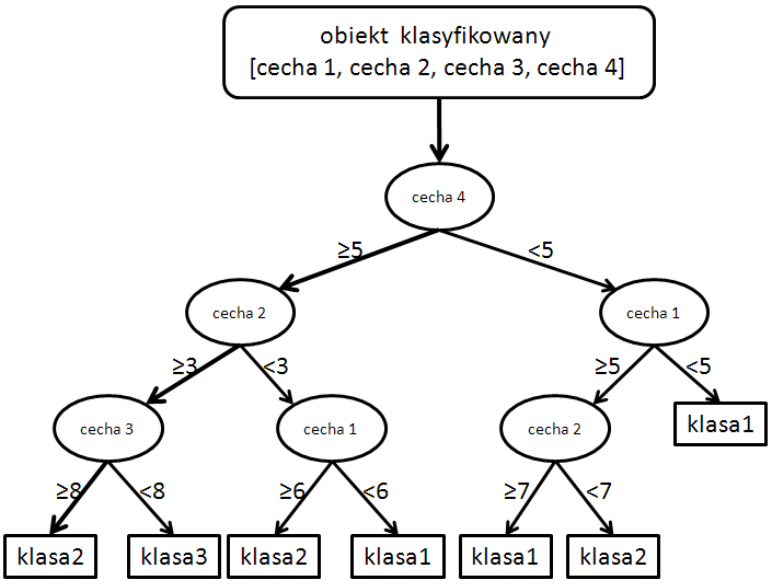
Rys. 2.3: Przepływ danych w klasyfikatorze gorliwym.

Przykład przepływu danych w klasyfikatorze gorliwym przedstawiono na Rysunku 2.3, gdzie $x_q^{(f)} \in X^F$ jest f-tą cechą q-tego obiektu, a $c_q \in C$ etykietą klasy tego obiektu. W przedstawionym na rysunku procesie klasyfikacji zbiór uczący stanowi Q obiektów opisanych przez F cech, z których każdy należy do jednej z klas zbioru C.

Z uwagi na tworzenie globalnego modelu klasyfikacji, poza nielicznymi wyjątkami (np. naiwny klasyfikator Bayesa), wektor danych testowych klasyfikatorów gorliwych musi zawierać informacje o wartościach wszystkich cech użytych do budowy modelu. Poniżej przedstawiono skrótowy opis kilku przykładowych rodzin klasyfikatorów gorliwych. Zostały one wybrane ze względu na wykorzystanie ich w badaniach porównawczych opisanych w dalszej części rozprawy.

Drzewa decyzyjne

Jednym z przykładów klasyfikatorów gorliwych są drzewa decyzyjne. Określenie to stosowane jest dla metody klasyfikacji obiektów, która bazuje na diagramach decyzyjnych zwanych drzewami. Drzewa decyzyjne, w celach poglądowych prezentowane są często w postaci graficznej jako graf acykliczny. W praktyce drzewa są strukturą danych reprezentowaną w pamięci komputera. Korzeń drzewa reprezentuje wszystkie obiekty, każdy wewnętrzny węzeł opisuje wybraną cechę, a liście drzewa reprezentują ostateczne przyporządkowanie danych do klas. Drzewa są konstruowane według zasady „dziel i zwyciężaj", która umożliwia podział problemu na podproblemy. Podział dokonywany jest w oparciu o test logiczny dla wybranej cechy (Rys. 2.4). Testem logicznym może być dowolny warunek lub funkcja zwracająca wartość logiczną.
W najprostszej postaci, ale oddającej zasadę podejmowania decyzji w drzewie decyzyjnym, mamy do czynienia z konstrukcją, która wymaga, aby w liściach znajdowały się elementy jednej klasy decyzyjnej. Wtedy liście opisują klasę decyzyjną, reprezentowaną przez przykłady ze zbioru treningowego. Ustalenie klasy następuje na podstawie testowania atrybutów. Podobnie postępujemy dla innych elementów drzewa decyzyjnego uzyskując ostatecznie rozpoznanie wszystkich klas.



Rys. 2.4: Drzewo decyzyjne.

Do najbardziej znanych klasyfikatorów opartych na koncepcji drzewa decyzyjnego zaliczyć można algorytm ID3 (ang. *Iterative Dychotomiser 3*) [48] oraz bazujące na nim algorytmy C4.5 [53] i CART (ang. *Classification and Regression Trees*) [8].
Podstawowymi zaletami algorytmów działających w oparciu o drzewa decyzyjne są:

a) szybka klasyfikacja,

b) przejrzysty opis procesu decyzyjnego,

c) możliwość podejmowania decyzji w oparciu o różne cechy na różnych poziomach drzewa.

Główną wadą tych metod są natomiast trudności w dostosowaniu modelu decyzyjnego do nowych obiektów uczących.

Klasyfikatory regułowe

Innym przykładem klasyfikatorów gorliwych są klasyfikatory regułowe. Opierają one swoje działanie o reguły decyzyjne w postaci:

$$\text{jeżeli } \varphi, \text{to } \psi, \tag{2.4}$$

gdzie φ jest testem logicznym (przesłanką), a ψ jest konkluzją reguły.

Istnieje wiele rodzajów reguł, mi.in.: logiczne, decyzyjne (bądź klasyfikacyjne), regresyjne, asocjacyjne, akcji, wzbraniające i inne [70]. Reguły decyzyjne używane są do opisu przykładów należących do tablicy decyzyjnej. Klasyfikator regułowy najczęściej korzysta z jednego z trzech schematów decyzyjnych: listy reguł, reguły o największym zaufaniu lub głosowania.
W fazie treningu generowany jest zbiór reguł Ω na podstawie tablicy decyzyjnej A. Oznaczmy ten zbiór Ω^A.
W fazie selekcji reguł w zbiorze Ω^A poszukiwany jest podzbiór wszystkich takich reguł, które wspierają obiekt $\mathbf{x}$. Oznaczmy ten podzbiór jako $D^A(\mathbf{x})$.
Faza klasyfikacji wyznacza klasę decyzyjną dla $\mathbf{x}$ według następującego schematu:
$D^A(\mathbf{x}) = \varnothing \rightarrow$ nie ma podstaw aby klasyfikować $\mathbf{x}$ do jakiejkolwiek klasy.
$D^A(\mathbf{x}) = c \rightarrow$ obiekt $\mathbf{x}$ należy do klasy c.
$D^A(\mathbf{x}) = \{c_1, \ldots, c_k\} \rightarrow$klasa, do której należy obiekt $\mathbf{x}$ jest wyznaczana na podstawie głosowania lub zaufania k reguł.

Szczególnym przypadkiem klasyfikatora regułowego jest klasyfikator 0-R, który jako wynik klasyfikacji zwraca identyfikator klasy najliczniej reprezentowanej w zbiorze treningowym, zatem jego jedyna reguła decyzyjna nie ma związku z wartościami cech wektora klasyfikowanego $\mathbf{x}$. Reguły podziału drzew decyzyjnych również mogą być wykorzystywane do budowy klasyfikatora regułowego, przykładem takiego klasyfikatora może być algorytm indukcji drzew decyzyjnych *C4.5* (*PART*) [16]. Dla przykładu z Rys. 2.4 możemy przedstawić ścieżkę decyzyjną w postaci złożonej reguły zapisaną jako Algorytm 1.

Algorytm 1: Przykładowa reguła decyzyjna.

```
1  begin
2      if (cecha4 ≥ 5) then
3          if (cecha2 ≥ 3) then
4              if (cecha3 ≥ 8) then
5                  return klasa2;
6              else
7                  return klasa3;
8              end
9          end
10     end
11 end
```

Klasyfikatory bayesowskie

Ostatnim z przedstawionych w niniejszej pracy przykładów klasyfikatorów gorliwych są klasyfikatory probabilistyczne, których typowym przykładem jest tzw. naiwny klasyfikator Bayesa. Jest to klasyfikator probabilistyczny bazujący na założeniu niezależności cech analizowanego obiektu (stąd określenie naiwny). Jeśli obiekty treningowe opisywane są wektorem cech i każdy obiekt treningowy posiada znaną *a priori* etykietę klasy $c_q \in C$ to przyjmuje się, że najbardziej prawdopodobną klasą, do której należy przypisać nowy, nieznany klasyfikowany obiekt, opisany wektorem cech $\mathbf{x} = \left[x^{(1)}, \ldots, x^{(F)}\right]$, jest klasa c, która maksymalizuje prawdopodobieństwo warunkowe *a priori* $P\left(c|x^{(1)}, \ldots, x^{(F)}\right)$. Przyjęto również, że liczba cech jest stała dla wszystkich obiektów, natomiast w poszczególnych obiektach, zarówno treningowych jak i klasyfikowanych, może brakować pewnych cech. Korzystając z twierdzenia Bayesa decyzję klasyfikatora *a posteriori* D_{map} wywieść można z prawdopodobieństwa *a priori*:

$$\begin{aligned} D_{map}(\mathbf{x}) &= \underset{c \in C}{\operatorname{argmax}}\, P\left(c|x^{(1)}, \ldots, x^{(F)}\right) \\ &= \underset{c \in C}{\operatorname{argmax}}\, \frac{P\left(x^{(1)}, \ldots, x^{(F)}|c\right)P(c)}{P\left(x^{(1)}, \ldots, x^{(F)}\right)} \\ &= \underset{c \in C}{\operatorname{argmax}}\, P\left(x^{(1)}, \ldots, x^{(F)}|c\right) P\left(c\right), \end{aligned} \tag{2.5}$$

gdzie $x^{(f)}$ to f-ta cecha z wektora $\mathbf{x}$, a $c \in C$ to klasa rozpatrywanego obiektu.

We wzorze (2.5) analizę mianownika można pominąć, gdyż $P\left(x^{(1)}, \ldots, x^{(F)}\right)$ nie zależy od c a tym samym nie ma wpływu na decyzję D_{map}. Prawdopodobieństwo $P\left(c\right)$ można estymować przez wyznaczenie stosunku liczby obiektów treningowych należących do klasy c do liczby

wszystkich obiektów treningowych.
Dzięki założeniu niezależności cech, prawdopodobieństwo warunkowe przynależności obiektu opisanego cechami $\left[x^{(1)},\ldots,x^{(F)}\right]$, do klasy c można wyrazić wzorem:

$$P\left(x^{(1)},\ldots,x^{(F)}|c\right) = \prod_{f=1}^{F} P\left(x^{(f)}|c\right), \tag{2.6}$$

co oznacza, że klasyfikacja dokonywana jest na podstawie prawdopodobieństwa przynależności pojedynczej cechy $x^{(f)}$ do klasy c. Prawdopodobieństwo $P\left(x^{(f)}|c\right)$ można szacować jako stosunek liczby obiektów treningowych z klasy c dla których wartość f-tej cechy jest równa $x^{(f)}$, do liczby wszystkich obiektów treningowych z klasy c. Prawdopodobieństwo warunkowe *a posteriori* przynależności rozpatrywanego przypadku $\mathbf{x} = \left[x^{(1)},\ldots,x^{(F)}\right]$ do klasy c można więc zapisać w postaci następującej formuły:

$$D_{NB}(\mathbf{x}) = \underset{c\in C}{\operatorname{argmax}}\, P(c) \prod_{f=1}^{F} P\left(x^{(f)}|c\right). \tag{2.7}$$

Do zalet naiwnego klasyfikatora bayesowskiego można zaliczyć prostą implementację, szybkość działania oraz możliwość określenia wartości funkcji wsparcia dla każdej z klas.
Do jego wad należy zaliczyć brak możliwości wizualizacji procesu klasyfikacji, a także trudność objaśnienia procesu podejmowania decyzji.

2.3 Zarys metod tworzenia komitetów klasyfikacyjnych

Klasyfikator komitetowy to grupa klasyfikatorów, których indywidualne decyzje są wykorzystywane do wyłonienia decyzji komitetu (np. na drodze głosowania), czyli do sklasyfikowania nowego obiektu [15]. Klasyfikatory komitetowe mogą wykonywać operację klasyfikacji szeregowo lub równolegle (niezależnie od siebie). W skład komitetu mogą wchodzić klasyfikatory jednego typu lub różnych typów. Możemy zatem mówić o komitetach klasyfikatorów homogenicznych (jednorodnych) i heterogenicznych (niejednorodnych). Jednorodne komitety klasyfikatorów mogą być budowane m.in. w oparciu o:

a) ***Próbkowanie danych treningowych*** - każdy z klasyfikatorów wchodzących w skład komitetu wykorzystuje w celu budowy modelu klasyfikacji inny podzbiór obiektów treningowych. Przykładem techniki budowy komitetów klasyfikacyjnych opartych o próbkowanie danych jest *Bagging* [17]. Algorytm w każdej iteracji losuje ze zwracaniem reprezentatywną próbkę przykładów ze zbioru danych uczących i w oparciu o wylosowane przykłady buduje model klasyfikacji. Powstaje szereg klasyfikatorów, które równolegle

klasyfikują ten sam obiekt. Decyzja o przypisaniu obiektu do klasy podejmowana jest w głosowaniu większościowym.

b) ***Podział przestrzeni cech danych treningowych*** - szczególny przypadek próbkowania danych treningowych. W metodzie tej wszystkie klasyfikatory wchodzące w skład komitetu wykorzystują do budowy modelu wszystkie obiekty treningowe, jednak każdy z nich wykorzystuje tylko pewien podzbiór cech obiektów treningowych [59].

c) ***Modyfikowanie etykiet klas danych treningowych*** - Metody tworzenia komitetów klasyfikacyjnych bazujące na modyfikowaniu etykiet klas danych treningowych często są wykorzystywane do dekompozycji klasyfikacji wieloklasowej na komitet klasyfikatorów binarnych. Najprostszą metodą tego typu jest metoda OPC (*ang.* One Per Class) zwana również *jeden przeciw wszystkim* [38].
Innym przykładem dekompozycji jest metoda ECOC (*ang.* Error-Correcting Output Codes) [58]. Jej idea wywodzi się z kodów korekcyjnych stosowanych do zapisu danych na nośnikach cyfrowych. W metodzie tej liczba klasyfikatorów wchodzących w skład komitetu jest z reguły większa niż liczba klas. Każdej klasie przypisywane jest l-bitowe słowo kodowe, natomiast komitet tworzy l klasyfikatorów binarnych, zwracających poszczególne bity słowa kodowego. Decyzja komitetu jest wyłaniana jako klasa, której słowo kodowe jest najbardziej podobne do otrzymanego ciągu bitów. Powszechnie używaną w tym przypadku miarą odległości jest odległość Hamminga [6].

Rozdział 3

Miary jakości klasyfikacji

Jakość klasyfikacji może być szacowana w oparciu o liczbę poprawnie rozpoznanych obiektów należących do rozpatrywanej klasy. W przypadku klasyfikacji binarnej, przez analogię do wyniku testu diagnostycznego, klasy zwykło się nazywać „pozytywną" i „negatywną". W takim przypadku można zdefiniować 4 podstawowe współczynniki tworzące tzw. *tabelę krzyżową* [65], która jest podstawą do oceny jakości klasyfikacji dwuklasowej. Wspomniana tabela zawiera cztery pola opisujące: liczbę obiektów *prawdziwie pozytywnych — pp*, liczbę obiektów *prawdziwie negatywnych — pn*, liczbę obiektów *fałszywie pozytywnych — fp* oraz liczbę obiektów *fałszywie negatywnych — fn*. Popularne miary jakości klasyfikacji binarnej wykorzystujące powyższe współczynniki przedstawiono w Tabeli 3.1.

Tabela 3.1: Współczynniki oceny jakości klasyfikacji binarnej.

Miara	Wzór	Opis
(a) *dokładność*	$\frac{pp + pn}{pp + fn + fp + pn}$	Stosunek sumy wyników prawdziwie pozytywnych i negatywnych do sumy wyników prawdziwie pozytywnych i negatywnych oraz fałszywie pozytywnych i negatywnych.
(b) *precyzja*	$\frac{pp}{pp + fp}$	Stosunek wyników prawdziwie pozytywnych do sumy prawdziwie pozytywnych i prawdziwie negatywnych.
(c) *czułość*	$\frac{pp}{pp + fn}$	Stosunek wyników prawdziwie pozytywnych do sumy prawdziwie pozytywnych i fałszywie negatywnych.
(d) *swoistość*	$\frac{pn}{fp + pn}$	Stosunek wyników prawdziwie negatywnych do sumy prawdziwie negatywnych i fałszywie pozytywnych.
(e) *F-Measure*	$\frac{2 \times precyzja \times czułość}{precyzja + czułość}$	Średnia harmoniczna czułości i precyzji.
(f) *G-Measure*	$\sqrt{precyzja \times czułość}$	Średnia geometryczna czułości i precyzji.
(g) *AUC*	$\frac{(czułość + swoistość)}{2}$	Pole powierzchni pod krzywą ROC [7].

W wielu pracach można spotkać opisy eksperymentów, w których jedynym podanym wynikiem klasyfikacji jest jej dokładność wyznaczona zgodnie ze wzorem (b) z Tabeli 3.1. Należy wtedy pamiętać o niedoskonałości tej miary, gdyż jej wynik jest silnie uzależniony od rozkładu klas w zbiorze testowym. Łatwo to wykazać: przykładowo, jeżeli w zbiorze testowym znajduje się 99 obiektów negatywnych (reprezentujących przypadki osób zdrowych) i 1 obiekt pozytywny (reprezentujący przypadek osoby chorej), to klasyfikator typu 0R, który zawsze zwraca etykietę klasy większościowej (w tym przypadku wynik negatywny), uzyska następujące wyniki klasyfikacji: $pn = 99$, $pp = 0$, $fn = 1$, $fp = 0$, co oznacza, że zgodnie z wzorem (a) z Tabeli 3.1 uzyska *dokładność* = 0,99. Jak widać ocena jakości klasyfikacji jedynie w oparciu o miarę *dokładności* może być dobra, pomimo źle działającego algorytmu klasyfikacji. Istnieją jednak miary oceny klasyfikacji, które wykażą niezdolność tego klasyfikatora do wykrywania obiektów należących do klasy mniejszościowej – na przykład wartości współczynników *F-Measure* jak i *G-Measure* (Tabela 3.1) dla opisanego przypadku będą wynosiły 0, co jest najgorszym możliwym wynikiem.

Wymienione w Tabeli 3.1 współczynniki stosuje się w klasyfikacji dwuklasowej, ale przy odpowiedniej interpretacji można je rozszerzyć na klasyfikatory L-klasowe. Powszechnym rozwiązaniem jest wyznaczanie wartości współczynników dla każdej z klas $c \in \{1, \ldots, L\}$ według schematu klasa *„pozytywna”* $c^+ = l$, klasa *„negatywna”* $c^- = \{1, \ldots, l-1, l+1, \ldots, L\}$, a następnie uśrednienie uzyskanych wartości. Wyznaczone w ten sposób współczynniki dla l-tej klasy będą oznaczane jako pp_l, pn_l, fp_l, fn_l. Niestety takie uśrednianie nie jest pozbawione wad – klasyfikacja *prawdziwie negatywna* może oznaczać obiekt klasyfikowany błędnie, gdyż *prawidłowo negatywny* oznacza, że obiekt został przypisany do którejkolwiek z „negatywnych” klas, niekoniecznie do tej, do której rzeczywiście należy. W literaturze można spotkać dwie metody uśredniania miar jakości klasyfikacji, określane mianem uśredniania w skali mikro (μ – lokalnie, co do przypadku) i w skali makro (M – globalnie, co do klasy) [46]. Uśrednianie to odnosi się zarówno do czterech podstawowych wielkości – liczby przypadków sklasyfikowanych *prawdziwie pozytywnie*, *prawdziwie negatywnie*, *fałszywie pozytywnie* i *fałszywie negatywnie*, jak i miar pochodnych. Jednak w przypadku miar pochodnych, w literaturze można spotkać różne ich definicje. Wybrane przykłady uśrednionych miar oceny jakości klasyfikacji zostały przedstawione w Tabeli 3.2.

Tabela 3.2: Współczynniki oceny jakości klasyfikacji L-klasowej.

Miara	Wzór	Opis
(a) *średnia dokładność*	$\frac{\sum_{l=1}^{L} \frac{pp_l + pn_l}{pp_l + fn_l + fp_l + pn_l}}{p}$	Średnia dokładność klasyfikatora dla wszystkich klas.
(b*) $precyzja_\mu$	$\frac{\sum_{l=1}^{L} pp_l}{\sum_{l=1}^{L} (pp_l + fp_l)}$	Lokalnie uśredniona precyzja dla wszystkich klas.
(c) $czułość_\mu$	$\frac{\sum_{l=1}^{L} pp_l}{\sum_{l=1}^{L} (pp_l + fn_l)}$	Lokalnie uśredniona czułość dla wszystkich klas.
(d) $precyzja_M$	$\frac{\sum_{l=1}^{L} \frac{pp_l}{pp_l + fp_l}}{p}$	Globalnie uśredniona precyzja dla wszystkich klas.
(e) $czułość_M$	$\frac{\sum_{l=1}^{L} \frac{pp_l}{pp_l + fn_l}}{p}$	Globalnie uśredniona czułość dla wszystkich klas.
(f) $F\text{-}Measure_\mu$	$\frac{2 \times precyzja_\mu \times czułość_\mu}{precyzja_\mu + czułość_\mu}$	Wartość *F-Measure* obliczona w oparciu o lokalnie uśrednione miary bazowe.
(g) $F\text{-}Measure_M$	$\frac{2 \times precyzja_M \times czułość_M}{precyzja_M + czułość_M}$	Wartość *F-Measure* obliczona w oparciu o globalnie uśrednione miary bazowe.
(h) $F\text{-}Measure_{M^*}$	$\frac{\sum_{l=1}^{L} \left(\frac{2 \times \frac{pp_l}{pp_l + fp_l} \times \frac{pp_l}{pp_l + fn_l}}{\frac{pp_l}{pp_l + fp_l} + \frac{pp_l}{pp_l + fn_l}} \right)}{p}$	Globalnie uśredniona wartość *F-Measure* (wariant alternatywny).
(i) $G\text{-}Measure_\mu$	$\sqrt{precyzja_\mu \times czułość_\mu}$	Wartość *G-Measure* obliczona w oparciu o lokalnie uśrednione miary bazowe.
(j) $G\text{-}Measure_M$	$\sqrt{precyzja_M \times czułość_M}$	Wartość *G-Measure* obliczona w oparciu o lokalnie uśrednione miary bazowe.

*) W artykułach naukowych i oprogramowaniu statystycznym można znaleźć różne definicje niektórych miar wtórnych, niekiedy autorzy wprowadzają własne miary lub własne nazwy dla istniejących miar, np. w środowisku *KNIME* istnieje miara *Overall Accuracy* (*ogólna dokładność* - ACC), która w rzeczywistości jest *lokalnie uśrednioną precyzją* [66].

Rozdział 4

Przegląd metod klasyfikacji danych z wadami

W rzeczywistych zbiorach danych mogą występować różnego rodzaje niedoskonałości utrudniające wykorzystanie tych zbiorów w procesie klasyfikacji. W niniejszym Rozdziale przedstawione zostały dwa przykłady zbiorów danych posiadających wady – zbiory niepełne oraz niezrównoważone. Wymienione zostały również, spotykane w literaturze, sposoby postępowania z takimi zbiorami danych w odniesieniu do klasyfikacji nadzorowanej.

4.1 Metody postępowania z danymi niepełnymi

Przez dane niepełne rozumie się dane, dla których jedna lub więcej ze składowych wektora opisującego cechy obiektu nie posiada przypisanej wartości (lub posiada wartość *null*), innymi słowy przestrzeń danych wektora niepełnego zawiera podzbiór cech przestrzeni wektora pełnego.

Dane niepełne mogą mieć różny charakter. W literaturze spotyka się następujące charakterystyki danych niepełnych [39, 54, 56]:

a) ***Dane brakujące nielosowo*** (*NMAR* - *ang.* Not Missing at Random) - dotyczą przypadku, gdy brak jest możliwości wykonania jakiegoś pomiaru lub badania na skutek niewystępowania danej cechy w badanym obiekcie. Informacja o brakujących cechach obiektu na podstawie których dokonywana będzie klasyfikacja jest wykorzystywana w procesie klasyfikacji.

b) ***Dane brakujące całkowicie losowo*** (*MCAR* - *ang.* Missing Completely at Random) - dane, dla których nie istnieje związek pomiędzy brakiem cech opisujących obiekt, a ich wartościami ani wartościami innych cech tego obiektu. Wartość tych cech istniała, ale nie została zmierzona.

c) ***Dane brakujące losowo*** (*MAR* - *ang.* Missing at Random) - nie wykazują bezpośredniego związku pomiędzy brakiem cechy opisującej obiekt, a jej wartością. Mogą jednak zależeć warunkowo od wartości innych cech tego obiektu. Wartość tych cech nie została zmierzona, ale może być przybliżona w oparciu o wartości innych, zmierzonych cech.

Sposób postępowania z danymi niepełnymi zależy od ich charakterystyki. W przypadku danych brakujących nielosowo, brak cechy jest informacją o stanie obiektu która może zostać wykorzystana w procesie klasyfikacji. Tego typu brak wartości cechy należy traktować jako odrębną wartość danej cechy (np. poprzez przypisanie jej unikalnej wartości). W przypadku danych brakujących całkowicie losowo, przy dostatecznej liczbie pełnych wektorów cech, wektory niepełne można usunąć ze zbioru. Możliwe jest również zastosowanie jednej z technik uzupełniania danych, których stosowanie jest konieczne w przypadku danych brakujących losowo.

W przypadku usuwania danych istnieją dwie możliwości:

a) usuwanie niepełnych wektorów cech,
b) redukcję cech, które nie są określone we wszystkich wektorach.

W zależności od rozkładu brakujących wartości w obiektach można stosować jedną z tych technik lub obie w powiązaniu ze sobą.

W przypadku uzupełniania danych wyróżnić można również dwie możliwości:

a) ***uzupełnianie jednokrotne*** – polega na uzupełnieniu brakującej cechy obiektu wartością obliczoną metodami statystyczni, jak np. średnia czy mediana. Wartości te mogą być obliczone globalnie, lub na podstawie wybranej grupy obiektów ze zbioru uczącego, np. obiektów należących do tej samej klasy co obiekt uzupełniany. Możliwie jest użycie bardziej złożonych algorytmów uzupełniania jednokrotnego, np. regresji lub metody *EM* (oczekiwania-maksymalizacji) [51, 54].

b) ***uzupełnienie wielokrotne*** – polega na wygenerowaniu kilku pełnych zbiorów danych różniących się wartościami uzupełnianych cech. Uzupełniana cecha w każdym z tych zbiorów ma inną wartość i może uwzględniać statystyczny rozkład wartości uzupełnianej cechy w całym zbiorze uczącym. Przykładem takiego algorytmu jest *EMB*, będący rozwinięciem algorytmu *EM* [28, 29]. Algorytm ten został zaimplementowany w pakiecie *Amelia* przeznaczonym dla środowiska *R*.

Sposoby uzupełniania danych mogą znacząco wpływać na jakość klasyfikacji, co pokazane będzie w dalszej części pracy. Autorska metoda klasyfikacji nie wymaga uzupełniania danych dla zachowania wysokiej skuteczności klasyfikacji obiektów, co zostało wykazane w trakcie badań porównawczych, których wyniki są prezentowane w dalszej części pracy.
Dane niepełne mogą się znajdować zarówno w zbiorze danych uczących, jak i w zbiorze danych klasyfikowanych. Możliwe są następujące warianty klasyfikacji z wykorzystaniem danych niepełnych:

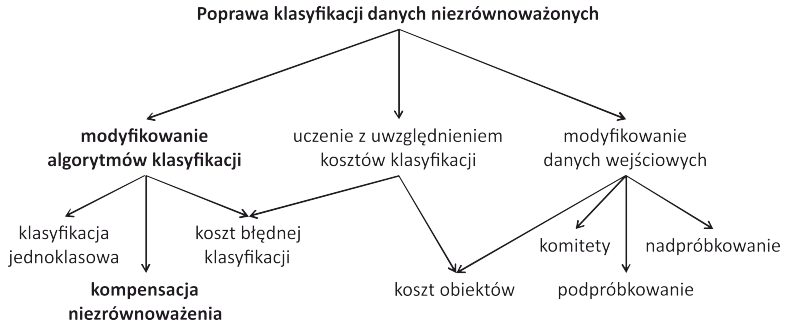
a) klasyfikator uczony na danych pełnych, klasyfikuje dane niepełne,

b) klasyfikator uczony na danych niepełnych, klasyfikuje dane pełne,
c) klasyfikator uczony na danych niepełnych, klasyfikuje dane niepełne.

Pierwszy z wymienionych wariantów klasyfikacji danych niepełnych nie stanowi problemu dla klasyfikatorów leniwych, gdyż reguły decyzyjne klasyfikatora budowane są dopiero w trakcie klasyfikacji obiektu. Oznacza to, że decyzja klasyfikatora może być podejmowana w oparciu o przestrzeń cech obiektu klasyfikowanego. W przypadku danych niepełnych w zbiorze uczącym, poza pewnymi wyjątkami, zachodzi potrzeba usunięcia lub uzupełniania niepełnych wektorów danych.

4.2 Metody postępowania z danymi niezrównoważonymi

Przez pojęcie danych niezrównoważonych rozumie się sytuację, gdy w zbiorze uczącym liczebność obiektów należących do poszczególnych klas jest zróżnicowana. Wpływ niezrównoważenia danych treningowych jest różny na różne klasyfikatory [43, 69]. Jako przykład klasyfikatora odpornego na niezrównoważenie klas można podać naiwny klasyfikator Bayesa, natomiast szczególnie wrażliwe na niezrównoważenie klas są klasyfikatory bazujące na entropii (regułowe, drzewa decyzyjne), oraz k-NN.
Istnieje wiele metod poprawiających jakość klasyfikacji danych niezrównoważonych [62]. Ich podział przedstawiono na Rys. 4.1.



Rys. 4.1: Wybrane metody poprawy jakości klasyfikacji danych niezrównoważonych.

Najbardziej oczywistą metodą jest tzw. podpróbkowanie obiektów klasy większościowej, czyli odrzucenie części obiektów z klasy lub klas większościowych, tak aby wyrównać liczności wszystkich klas w zbiorze treningowym [62]. Konsekwencją tej metody jest utrata części informacji.

Metodą podobną do podpróbkowania obiektów klasy większościowej jest metoda polegająca na tworzeniu komitetów klasyfikatorów z podpróbkowaniem danych uczących [52, 72]. Na podstawie niezrównoważonego zbioru danych uczących, w oparciu o próbkowanie obiektów,

tworzona jest pula zrównoważonych podzbiorów uczących. Podzbiory te są używane do trenowania klasyfikatorów wchodzących w skład komitetu. W wyniku takiego próbkowania pewna liczba obiektów z klasy większościowej zostanie odrzucona.

Przeciwnym podejściem jest nadpróbkowanie, czyli powielanie lub generowanie obiektów sztucznych należących do klasy mniejszościowej [68]. Technika ta niesie ze sobą ryzyko przeuczenia klasyfikatora, czyli zbyt ścisłego dopasowania klasyfikatora do danych uczących.

Powyższe metody bazują na modyfikowaniu danych, ale możliwe jest dostosowanie algorytmów klasyfikacji tak, aby pracowały one w oparciu o niezrównoważone zbiory treningowe. Jedną z takich metod jest użycie komitetów klasyfikatorów jednoklasowych, czyli trenowanych wyłącznie w oparciu o obiekty należące do klasy, którą mają rozpoznawać [14, 73]. Innym możliwym podejściem jest projektowanie algorytmów z modelem decyzyjnym klasyfikatora, niewrażliwym na niezrównoważenie klas w zbiorze uczącym [17]. Takie podejście będzie stosowane w niniejszej pracy.

Niektóre źródła zaliczają również do metod poprawy jakości klasyfikacji niezrównoważonych danych uczenie z uwzględnieniem kosztów klasyfikacji (*ang.* Cost-Sensitive Learning) [61, 63]. Metoda ta polega na ustaleniu współczynników kosztów błędnej klasyfikacji obiektów lub klas. Opisane powyżej metody postępowania z danymi niezrównoważonymi w procesie klasyfikacji obejmują również walidację krzyżową w oparciu o niezrównoważony zbiór treningowy.

Rozdział 5

Przegląd metod selekcji cech

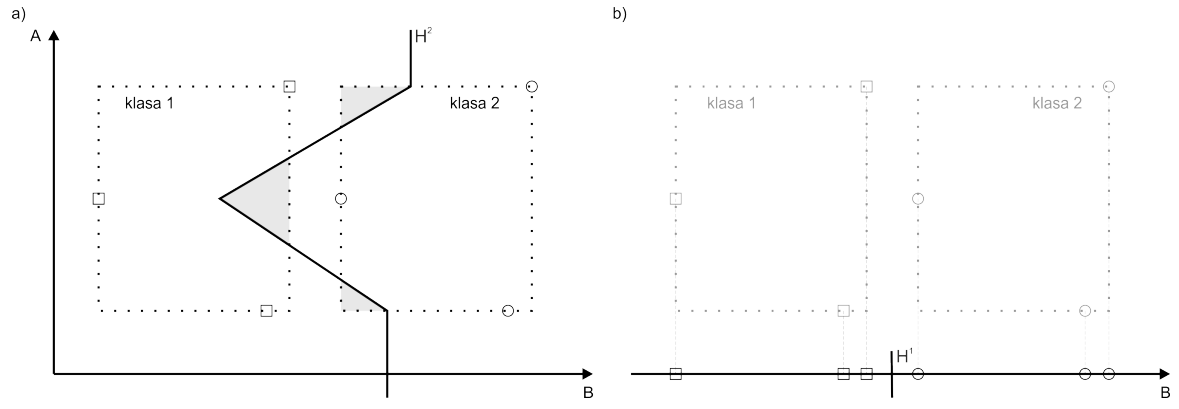
Ważnym zadaniem związanym z klasyfikacją obiektów jest odpowiednia redukcja danych. Redukcja danych może polegać na [21, 74]:

a) selekcji informacji, czyli redukcji liczby cech opisujących obiekt,

b) zastąpieniu ciągłego zakresu zmienności cechy wartościami dyskretnymi,

c) redukcji liczby obiektów reprezentujących poszczególne klasy.

Największe znaczenie ma selekcja rozumiana jako wybór cech o najlepszych właściwościach dyskryminujących, które zostaną wykorzystane w procesie klasyfikacji obiektów. Nie jest to problem trywialny, ponieważ kryteria doboru metody selekcji cech są ściśle związane z metodą klasyfikacji danych.
Za przykład może posłużyć klasyfikator minimalno-odległościowy. Jego zadaniem jest znalezienie hiperpowierzchni rozgraniczającej obiekty należące do różnych klas. Jeśli przyjąć, że przestrzeń cech jest zdefiniowana przez dwie cechy, to hiperpłaszczyzna ta będzie linią łamaną przebiegająca w równej odległości od punktów reprezentujących obiekty referencyjne [32]. Na Rys. 5.1 oznaczono te cechy jako A oraz B, zaś hiperpłaszczyznę rozgraniczającą obszary klas jako H^2. Obszary klas (wyznaczone przez obiekty referencyjne w przestrzeni $\mathbb{R}^2$ i $\mathbb{R}^1$) oznaczono linią przerywaną.



Rys. 5.1: Granice klas w przestrzeniach $\mathbb{R}^2$ oraz $\mathbb{R}^1$.

Jak wynika z Rys. 5.1a) taki klasyfikator nieprawidłowo sklasyfikował trzy fragmenty przestrzeni (obszary szare): dwa fragmenty przestrzeni zaklasyfikowane do klasy 1 (leżące na lewo

od H^2) leżą w obszarze klasy 2, zaś jeden fragment przestrzeni zaklasyfikowany do klasy 2 (leżący na prawo od H^2) leży w obszarze klasy 1. Gdyby jednak odrzucić cechę A i rzutować wszystkie obiekty na oś cechy B, hiperpłaszczyzna rozgraniczająca przestrzeń klas przyjęłaby formę prostej H^1, która optymalnie rozgranicza obszary klas 1 i 2 – Rys. 5.1b). Można zatem stwierdzić, że w przedstawionym przypadku odrzucenie cechy A przyczyniło się do poprawy jakości klasyfikacji. W wielowymiarowych zbiorach danych często tylko niektóre cechy wykazują związek z klasą obiektów, natomiast pozostałe cechy mogą być obojętne lub mogą zakłócać proces klasyfikacji.
Celem procesu selekcji cech może być zatem znalezienie podzbioru cech zapewniającego możliwie najlepszą dokładność klasyfikacji, bądź znalezienie minimalnego zbioru cech, którego dalsze rozszerzanie znacząco nie wpływa na poprawę jakości klasyfikacji. Metody selekcji nie gwarantują znalezienia optymalnego zbioru cech, zatem zbiór cech wyłoniony w procesie selekcji cech, nazywany jest sub-optymalnym.

W literaturze można spotkać różne kryteria podziału metod selekcji cech, jednak najczęściej spotykany jest następujący podział [60, 74]:

a) metody rankingowe (zwane także filtrami lub metodami filtracyjnymi),

b) metody opakowane (zwane także wrapperami),

c) metody wbudowane.

5.1 Metody rankingowe

Metody rankingowe bazują na niezależnej ocenie cech z wykorzystaniem ogólnych miar charakteryzujących dane. Może to być współczynnik korelacji między wartościami cech a przynależnością do określonej klasy, np. współczynnik Pearsona [11]. Rankingi budowane są też w oparciu o miarę zysku informacyjnego (*ang.* Information Gain) i względny zysk informacyjny (*ang.* Gain Ratio) [25]. Do rankinkowych metod selekcji cech zalicza się również procedury ReliefF [35] i CFS [23, 50]. Zbiór cech obiektu jest poddawany filtracji opartej o wcześniej zbudowany ranking cech w celu określenia potencjalnie najlepszego podzbioru cech przed rozpoczęciem budowy modelu klasyfikacyjnego [71].

5.2 Metody opakowane

Metody opakowane oceniają poszczególne podzbiory cech z wykorzystaniem algorytmu uczenia maszynowego, który docelowo będzie użyty do klasyfikacji danych. Algorytm uczący jest w tym

przypadku elementem procedury selekcji cech, a do oszacowania dokładności klasyfikatora korzystającego z określonego podzbioru cech wykorzystywana jest walidacja krzyżowa i jedna z miar oceny jakości klasyfikacji. Wyłanianie podzbioru cech może odbywać się na różne sposoby. Typowymi strategiami są: przeszukiwanie w przód, przeszukiwanie wstecz oraz tworzenie indywidualnego rankingu.
Na potrzeby opisu przyjmujemy, że zbiór danych uczących ma formę macierzy $\mathbf{X}$:

$$\mathbf{X} = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(F)} \\ x_2^{(1)} & x_2^{(2)} & \cdots & x_2^{(F)} \\ \vdots & \vdots & \ddots & \vdots \\ x_Q^{(1)} & x_Q^{(2)} & \cdots & x_Q^{(F)} \end{bmatrix}. \tag{5.1}$$

W powyższym zapisie $x_q^{(f)}$ oznacza wartość f-tej cechy z q-tego wektora uczącego. Z macierzą danych uczących skojarzony jest wektor klas $\mathbf{c}$:

$$\mathbf{c} = \left[c_1, c_2, \ldots, c_Q\right]^T. \tag{5.2}$$

Dla poprawienia czytelności zapisu wprowadzone zostało pojęcie wektora cechy $\mathbf{x}^{(f)}$, który należy rozumieć jako transpozycję f-tej kolumny macierzy $\mathbf{X}$:

$$\mathbf{x}^{(f)} = \left[x_1^{(f)}, x_2^{(f)}, \ldots, x_Q^{(f)}\right]^T, \quad f \in \{1, 2, \ldots, F\}. \tag{5.3}$$

Procedura przeszukiwania w przód

Procedura przeszukiwania w przód (Algorytm 2) rozpoczyna się przy pustym podzbiorze wybranych cech. W pierwszym kroku wyznaczana jest jakość klasyfikacji zbioru $\mathbf{X}$ w oparciu o pojedynczą cechę f przy użyciu algorytmu klasyfikacji D. Możliwe jest stosowanie różnych odmian walidacji krzyżowej ($\mathit{WalidacjaKrzyżowa}(D, [\mathbf{X}_{test}\mathbf{c}])$ w linii 6 Algorytmu 2) i różnych miar oceny jakość klasyfikacji. W ten sposób testowana jest każda z F cech ze zbioru uczącego, w celu wyłonienia najlepszej i dołączenia jej do podzbioru wybranych cech $\mathbf{X}_{opt}$. W kolejnym kroku testowane są wszystkie kombinacje cechy wybranej w poprzednim kroku z jedną z dotychczas niewybranych cech. Do podzbioru $\mathbf{X}_{opt}$ dołączana jest cecha, której dołączenie pozwoliło na uzyskanie najlepszego jakości klasyfikacji. Procedura ta jest powtarzana aż do osiągnięcia kryterium stopu (KS w linii 12 algorytmie 2). Kryterium stopu (KS) może sprowadzać się do określenia docelowej liczby cech, określonej jakości klasyfikacji lub braku poprawy jakości klasyfikacji w kolejnej iteracji algorytmu.
Kolejne kolumny Tabeli 5.1 reprezentują iteracje Algorytmu 2. Pogrubioną czcionką oznaczono zestaw cech wybrany w każdej iteracji. Komórki tabeli zawierają numery f cech $\mathbf{x}^{(f)}$ użytych do klasyfikacji oraz uzyskaną ogólną dokładność klasyfikacji (ACC). W przedstawionym przykła-

dzie największą dokładność klasyfikacji uzyskano w 4-tej iteracji w oparciu o czteroelementowy zbiór cech.

Algorytm 2: Procedura przeszukiwania w przód

Input:

- $\mathbf{X}$: zbiór danych treningowych
- $\mathbf{c}$: wektor klas
- F: liczba cech
- D: algorytm klasyfikacji
- KS: kryterium stopu

Output:

- $\mathbf{X}_{opt}$: wynikowy podzbiór cech

```
1  begin
2      X_opt ← {∅};
3      repeat
4          for f := 1 to F do
5              X_test ← X_opt ∪ {x^(f)};
6              ACC[f] := WalidacjaKrzyżowa(D, {X_test c});
               // ACC[f] : jednowymiarowa tablica wyników jakości klasyfikacji
7          end
8          f' := argmax_f (ACC[f]);
           // f' : indeks cechy dla której dokładność klasyfikacji była najlepsza
9          X_opt ← X_opt ∪ {x^(f')};
10         X ← X \ {x^(f')};
11         F := F − 1;
12     until (F = 0 ∨ KS = true);
13     return X_opt;
14 end
```

Tabela 5.1: Przykład działania algorytmu przeszukiwania w przód

Numer iteracji				
1	2	3	4	5
(1) 0,50	(1,4) 0,80	(1,3,4) 0,80	**(1,3,4,5) 0,90**	**(1,2,3,4,5) 0,80**
(2) 0,70	(2,4) 0,50	(2,3,4) 0,75	(2,3,4,5) 0,85	–
(3) 0,60	**(3,4) 0,85**	**(3,4,5) 0,90**	–	–
(4) 0,80	(4,5) 0,70	–	–	–
(5) 0,75	–	–	–	–

Procedura przeszukiwania wstecz

Procedura przeszukiwania wstecz (Algorytm 3) przebiega analogicznie, z tą różnicą, że rozpoczyna się od pierwotnego zbioru cech, z którego kolejno usuwane są cechy. Kryterium wyboru cechy do usunięcia w każdym kroku jest maksymalizacja miary jakości klasyfikacji.

Algorytm 3: Procedura przeszukiwania wstecz

Input:

$\mathbf{X}$:	zbiór danych treningowych
$\mathbf{c}$:	wektor klas
F:	liczba cech
D:	algorytm klasyfikacji
KS:	kryterium stopu

Output:

$\mathbf{X}_{opt}$:	wynikowy podzbiór cech

```
1  begin
2      repeat
3          for f := 1 to F do
4              X_test ← X \ {x^(f)};
5              ACC[f] := WalidacjaKrzyżowa(D, {X_test c});
               // ACC[f] : jednowymiarowa tablica wyników jakości klasyfikacji
6          end
7          f' := argmax_f (ACC[f]);
           // f' : indeks cechy dla której dokładność klasyfikacji była najlepsza
8          X ← X \ {x^(f')};
9          F := F − 1;
10     until (F = 0 ∨ KS = true);
11     X_opt ← X;
12     return X_opt;
13 end
```

Tabela 5.2: Przykład działania algorytmu przeszukiwania wstecz

Numer iteracji				
1	2	3	4	5
(1,2,3,4,5) 0,80	(2,3,4,5) 0,85	**(3,4,5) 0,90**	(3,4) 0,85	(3) 0,60
–	**(1,3,4,5) 0,90**	(1,4,5) 0,85	**(3,5) 0,95**	**(5) 0,75**
–	(1,2,4,5) 0,80	(1,3,5) 0,75	(4,5) 0,70	–
–	(1,2,3,5) 0,75	(1,3,4) 0,80	–	–
–	(1,2,3,4) 0,85	–	–	–

W Tabeli 5.2 najlepszy wynik klasyfikacji został uzyskany w 4-tej iteracji w oparciu o podzbiór 2 cech. Algorytm przeszukiwania wstecz testował inne kombinacje cech niż algorytm przeszukiwania w przód.

Procedura indywidualnego rankingu

Metoda indywidualnego rankingu cech (Algorytm 4) polega na jednorazowym zbudowaniu rankingu cech z uwzględnieniem jakości klasyfikacji bazującej na pojedynczych cechach. Pod tym względem przypomina ona metody rankingowe. Pozycja cechy w rankingu jest określana na podstawie walidacji krzyżowej i wybranej miary jakości. Metoda ta, podobnie jak naiwny klasyfikator Bayesa, zakłada niezależność cech. Do podzbioru sub-optymalnych cech dodawane są kolejno najlepsze cechy, według wcześniej przygotowanego rankingu, aż do osiągnięcia kryterium stopu. Niektóre źródła określają właśnie tę metodę mianem przeszukiwania w przód, zwłaszcza jeśli kryterium stopu (KS) zawiera dodatkowy etap krzyżowej walidacji i oceny jakości klasyfikacji nowo-powstałego podzbioru danych z uwzględnieniem historii wyników. W takim przypadku kryterium stopu może mieć formę warunku „**jeżeli** *jakość klasyfikacji* **nie wzrosła od** *poprzedniej iteracji* (lub od poprzednich 2-3 iteracji) **przerwij algorytm** i **zwróć** *zbiór danych oparty o najlepiej oceniony podzbiór cech*". Proponowana w rozprawie metoda selekcji cech bazuje na tym właśnie algorytmie.

Tabela 5.3: Przykład działania algorytmu indywidualnego rankingu

Numer iteracji				
1	2	3	4	5
(1) 0,50	**(4,5) 0,70**	**(2,4,5) 0,90**	**(2,3,4,5) 0,85**	**(1,2,3,4,5) 0,80**
(2) 0,70	–	–	–	–
(3) 0,60	–	–	–	–
(4) 0,80	–	–	–	–
(5) 0,75	–	–	–	–

Jak wynika z przykładów przedstawionych w Tabelach 5.1 – 5.3 żaden z zaprezentowanych algorytmów poszukiwania sub-optymalnego zbioru cech nie testuje wszystkich kombinacji cech, a zatem nie daje gwarancji znalezienie optymalnego zbioru cech. Cechą wspólną wymienionych metod jest liczba iteracji, która jest równa liczbie cech.

Najszybszą z przedstawionych metod jest metoda indywidualnego rankingu. Jej pierwsza iteracja jest identyczna jak w metodzie przeszukiwania w przód, jednak w kolejnych etapach sprawdzane są tylko kolejne najlepsze cechy z pierwszej iteracji. Niestety takie rozwiązanie nie zawsze gwarantuje dobry dobór cech. Lepszymi metodami selekcji są metody opakowane. Wyznaczane za pomocą metod opakowanych cechy pozwalają na uzyskanie lepszej jakość klasyfikacji, kosztem czasu traconego na selekcję cech.

Algorytm 4: Procedura indywidualnego rankingu

Input:

$\mathbf{X}$:	zbiór danych treningowych
$\mathbf{c}$:	wektor klas
F:	liczba cech
D:	algorytm klasyfikacji
KS:	kryterium stopu

Output:

$\mathbf{X}_{opt}$:	wynikowy podzbiór cech

begin
  $\mathbf{X}_{opt} \leftarrow \{\varnothing\}$;
  for f := 1 **to** F **do**
    $\mathbf{ACC}[f]$:= $WalidacjaKrzyżowa(D, \left\{\mathbf{x}^{(f)}\mathbf{c}\right\})$;
    `// ACC[`f`] : jednowymiarowa tablica wyników jakości klasyfikacji`
  end
  fc := 1;
  repeat
    f' := $\underset{f}{\operatorname{argmax}}\,(\mathbf{ACC}[f])$;
    `//` f' `: indeks cechy dla której dokładność klasyfikacji była najlepsza`
    $\mathbf{X}_{opt} \leftarrow \mathbf{X}_{opt} \cup \left\{\mathbf{x}^{(f')}\right\}$;
    $\mathbf{ACC}[f']$:= 0;
    fc := fc + 1;
  until ($fc = F \vee KS = true$);
  return F_{opt};
end

5.3 Metody wbudowane

Metody wbudowane stanowią wewnętrzny element algorytmu klasyfikacji, który na etapie tworzenia modelu decyzyjnego przypisuje poszczególnym cechom wagi bądź dokonuje ich eliminacji. W literaturze znaleźć można wiele algorytmów klasyfikacji z wbudowanym mechanizmem selekcji cech. Typowym przykładem wbudowanej selekcji cech są algorytmy *LASSO* [64] oraz *RIDGE* [42]. Są to tzw. metody regularyzacyjne lub regularyzowanej regresji liniowej. Niektórzy autorzy zaliczają też do tej grupy algorytmy redukcji drzew decyzyjnych, sieci neuronowe [19], metodę wektorów nośnych (*SVM*) [10, 40] czy analizę składowych głównych (*PCA*), choć ten ostatni algorytm jest raczej przykładem ekstrakcji cech, gdyż tworzy nową przestrzeń cech. Dużą zaletą metod wbudowanych jest ich szybkość. Metody te są wbudowane w klasyfikator, więc ich użycie nie wiąże się z dodatkowymi operacjami na zbiorze uczącym. Są częścią konkretnego algorytmu klasyfikacji i uwzględniają jego charakterystykę, co pozwala na uzyskanie dobrych wyników klasyfikacji. Można to uznać zarówno jako zaletę jak i wadę, gdyż jednocześnie czyni to metody wbudowane nieprzydatnymi dla innych algorytmów klasyfikacji niż te, w które zostały wbudowane.

Istnieje wiele tego typu metod, bardzo między sobą się różniących, a ich opisanie jest niemożliwe bez szczegółowego przedstawienia zasady działania klasyfikatorów w które są wbudowane. Zagadnienia te nie są jednak tematyką rozważaną w niniejszej rozprawie.

Rozdział 6

Metoda k najbliższych sąsiadów (k-NN)

Jak wspomniano w Podrozdziale 2.1 metoda k najbliższych sąsiadów (*ang.* k-Nearest Neighbors) zaliczana jest do klasyfikatorów leniwych. W metodzie tej obiekt $\mathbf{x}$ poddawany procesowi klasyfikacji, zaliczany jest do klasy c, do której należy większość spośród k obiektów $\mathbf{x}_q$ ze zbioru referencyjnego A, które są najbardziej podobne do $\mathbf{x}$ pod względem przyjętej miary podobieństwa. Najprostszym klasyfikatorem tego typu jest klasyfikator 1-NN, który jako decyzję zwraca klasę najbardziej podobnego obiektu ze zbioru referencyjnego [36].
Działanie klasyfikatora k-NN można opisać w następujący sposób. Niech:

$$A = \{(\mathbf{x}_q, c_q)\}_{q=1}^{Q}, \quad \mathbf{x}_q \in \mathbb{R}^F, \tag{6.1}$$

będzie F wymiarowym zbiorem zawierającym Q obiektów referencyjnych, a etykieta klasy pojedynczego obiektu ze zbioru referencyjnego $\mathbf{x}_q$ będzie oznaczona jako c_q.
Dla klasyfikowanego wielowymiarowego obiektu $\mathbf{x}$, znajdowany jest podzbiór $\bar{A}$ składający się z k najbliższych sąsiadów, z dostępnych obiektów zbioru referencyjnego A. W tym celu konieczne jest uporządkowanie obiektów $\mathbf{x}_q$ ze zbioru referencyjnego A zgodnie z ich odległością ρ (Tabela 6.1) od obiektu klasyfikowanego $\mathbf{x}$. Niech obiekty zbioru $\bar{A}$ mają oznaczenie $\bar{\mathbf{x}}_q$, a ich klasy $\bar{c}_q \in C$, wtedy:

$$A \supset \bar{A} = \{(\bar{\mathbf{x}}_q, \bar{c}_q)\}_{q=1}^{k}, \quad \bar{\mathbf{x}}_q \in \mathbb{R}^F. \tag{6.2}$$

Oznacza to, że przewidywana klasa $D_{kNN}(\mathbf{x})$ klasyfikowanego obiektu $\mathbf{x}$ może być określona na podstawie większościowego głosowania klas $\bar{c}_q$ najbliższych sąsiadów $\bar{\mathbf{x}}_q$, $q = 1, \ldots, k$ z podzbioru referencyjnego $\bar{A}$:

$$D_{kNN}(\mathbf{x}) = \underset{c \in C}{\operatorname{argmax}} \sum_{q=1}^{k} \left(I(\bar{c}_q = c) \right). \tag{6.3}$$

W przypadku danych pochodzących z przestrzeni $\mathbb{R}$ jako miarę podobieństwa zwykle stosuje się jedną z miar odległości opisanych w Tabeli 6.1.

Dane referencyjne oraz klasyfikowane powinny zostać poddane normalizacji lub standaryzacji. Ten zabieg gwarantuje, że wszystkie wymiary przestrzeni danych w których obliczana jest odległość mają jednakową istotność. W wyniku normalizacji danych otrzymujemy wektory $\hat{x}$, których wartości składowe są zawarte w przedziale $[0,1]$. Przyjmując, że przez $x_q^{(f)}$ oznaczamy

Tabela 6.1: Miary odległości stosowane w klasyfikatorze k-NN.

Opis	Miara odległości
Odległość euklidesowa	$\rho_e(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{f=1}^{F}\left(x^{(f)}-y^{(f)}\right)^2}$
Odległość Manhattan	$\rho_m(\mathbf{x},\mathbf{y}) = \sum_{f=1}^{F}\left\|x^{(f)}-y^{(f)}\right\|$
Odległość Czebyszewa	$\rho_c(\mathbf{x},\mathbf{y}) = \max_{f=1,\ldots,F}\left(\left\|x^{(f)}-y^{(f)}\right\|\right)$

wartość f-tej cechy z q-tego wektora referencyjnego, możemy zapisać:

$$\hat{x}^{(f)} = \frac{x^{(f)} - \min\limits_{q=1,\ldots,Q}\{x_q^{(f)}\}}{\max\limits_{q=1,\ldots,Q}\{x_q^{(f)}\} - \min\limits_{q=1,\ldots,Q}\{x_q^{(f)}\}}. \tag{6.4}$$

Standaryzacja polega na doprowadzeniu do stanu, w którym wartość średnia każdej z cech wynosi 0, a odchylenie standardowe wartości cechy w zbiorze danych wynosi 1:

$$\hat{x}^{(f)} = \frac{x^{(f)} - \text{mean}\{x_q^{(f)}\}}{\text{std}\{x_q^{(f)}\}}, q = 1,\ldots,Q. \tag{6.5}$$

Zamiast standaryzacji lub normalizacji danych możliwe jest użycie ważonej odległości Euklidesowej uwzględniającej odchylenia standardowe poszczególnych cech lub odległości Mahalanobisa używającej jako wagi estymatora macierzy kowariancji. W przypadku danych jakościowych, miarą niepodobieństwa obiektów, reprezentowanych wektorem cech, może być np. współczynnik Sneatha:

$$\rho(\mathbf{x},\mathbf{y}) = \frac{1}{F}\sum_{f=1}^{F} I\left(x^{(f)} \neq y^{(f)}\right), \tag{6.6}$$

gdzie I jest funkcją indykatorową, a $\mathbf{x} = [x^{(1)},\ldots,x^{(F)}]$ oraz $\mathbf{y} = [y^{(1)},\ldots,y^{(F)}]$ są wektorami cech obiektów $x^{(f)} \in X$ oraz $y^{(f)} \in X$.
Jest to tzw. miara dopasowania obiektów. Innymi wykorzystywanymi miarami niepodobieństwa obiektów mogą być na przykład statystyka χ^2 oraz współczynnik Jaccarda. Kompleksowy przegląd miar odległości/podobieństwa można znaleźć w [9].

6.1 Modyfikacje metody k-NN bazujące na wagach

Istnieje wiele modyfikacji oryginalnej metody k-NN. Ich szczegółowe przeznaczenie jest różne ale wszystkie mają za zadanie poprawiać dokładność klasyfikacji oryginalnej metody. Niektóre z nich zostały opisane poniżej. Metody te będą wykorzystywane w badaniach porównawczych

z klasyfikatorami opracowanymi na potrzeby niniejszej pracy.
Wiele modyfikacji metody k-NN bazuje na współczynnikach wagowych przypisywanych do obiektów uczących umożliwiających minimalizację wpływu doboru parametru k. Realizowane jest to przez uwzględnienie odległości obiektu referencyjnego od klasyfikowanego w procesie wyłaniania decyzji klasyfikatora. Poniżej przedstawione zostały najbardziej reprezentatywne metody pozwalające na poprawę własności klasyfikatorów k-NN w oparciu o współczynniki wagowe.

Dla wszystkich prezentowanych tu modyfikacji ważona decyzja klasyfikatora k-NN przyjmuje postać:

$$D_{kNN}(\mathbf{x}) = \underset{c \in C}{\operatorname{argmax}} \sum_{q=1}^{k} \left(w_q \cdot I(\bar{c}_q = c) \right), \tag{6.7}$$

gdzie w_q jest wagą skojarzoną z q-tym obiektem zbioru referencyjnego. Dobór wag w_q może być różny, co przedstawiono poniżej.

Metoda Wk-NN

Jedną z metod pozwalających na zmniejszenie istotności wpływu doboru parametry k na jakość klasyfikacji polegających na wprowadzeniu dodatkowych wag uwzględniających odległość obiektu analizowanego $\mathbf{x}$ od obiektu referencyjnego $\mathbf{x}_q$ jest metoda oznaczana w literaturze jako Wk-NN (*ang.* Weighted k-NN) [18, 20].

W metodzie Wk-NN zakłada się, że obiekty referencyjne znajdujące się bliżej (w sensie odległości) od obiektu klasyfikowanego powinny mieć większy wpływ na decyzję klasyfikatora, niż obiekty odległe. Istnieją różne warianty metody Wk-NN wykorzystujące współczynniki wagowe oparte o odległości jednak dla niewielkich wartości k generują one zbliżone wyniki [18, 20, 30]. Waga w_q dla q-tego sąsiada klasyfikowanego obiektu $\mathbf{x}$ może być zdefiniowana na kilka sposobów, najprostszą z formuł jest:

$$w_q = 1 - \rho(\mathbf{x}, \mathbf{x}_q). \tag{6.8}$$

Wzór (6.8) zakłada, że odległości $\rho(\mathbf{x}, \mathbf{x}_q)$ są znormalizowane i mieszczą się w przedziale $[0, 1]$. Bez tego założenia konieczna jest normalizacja wag. W tym celu z każdym elementem zbioru referencyjnego można skojarzyć odległość $d_q = \rho(\mathbf{x}, \mathbf{x}_q)$ klasyfikowanego obiektu $\mathbf{x}$ od elementów $\mathbf{x}_q$ ze zbioru referencyjnego, gdzie ρ może być jedną z miar odległości z Tabeli 6.1.
Niech $d_{min} = \min\{d_q\}_{q=1}^{Q}$ oraz $d_{max} = \max\{d_q\}_{q=1}^{Q}$ oznaczają najmniejszą i największą odległość klasyfikowanego obiektu $\mathbf{x}$ od elementów $\mathbf{x}_q$ zbioru referencyjnego. Dla takich założeń wagę w_q

można wyznaczyć na podstawie formuły:

$$w_q = \begin{cases} \dfrac{d_{max} - d_q}{d_{max} - d_{min}} & \text{dla } d_{max} \neq d_{min} \\ 1 & \text{dla } d_{max} = d_{min} \end{cases} . \tag{6.9}$$

W literaturze można spotkać dwie odmiany metody Wk-NN:

a) Zgodnie z definicją używaną w środowisku `WEKA` [24], wagi w_q są przypisane do każdego obiektu $\mathbf{x}_q$ ze zbioru referencyjnego. W przypadku dużych zbiorów referencyjnych i stosunkowo niewielkiej liczby rozpatrywanych najbliższych sąsiadów, może to skutkować przypisaniem im bardzo zbliżonych wag. Metoda ta będzie oznaczana jako Wk-NN^g.

b) W pracach [20, 30] autorzy proponują wyznaczenie wag jedynie dla k najbliższych sąsiadów. Wagi są obliczane tak samo jak w formule (6.9), lecz teraz odległości d_{min} oraz d_{max} oznaczają odpowiednio najbliższy i najbardziej odległy obiekt spośród k najbliższych sąsiadów. W konsekwencji k-ty (leżący najdalej) sąsiad ma wagę $w_k = 0$, co oznacza, że w ogóle nie jest brany pod uwagę przy podejmowaniu decyzji przez klasyfikator. Równocześnie wagi rozpatrywanych k najbliższych sąsiadów są bardziej zróżnicowane. Metoda ta będzie oznaczana jako Wk-NN^l.

Metoda DWk-NN

W pracy [20] autorzy zaproponowali wagę, mającą jeszcze mocniej ograniczyć wpływ doboru parametru k na jakość klasyfikacji. Metoda ta będzie oznaczana jako DWk-NN (*ang.* Dual Distance Weighted k-NN). W tej metodzie waga w_q obliczana jest na podstawie dwóch ilorazów. Pierwszy jest taki sam jak w metodzie Wk-NN (wzór 6.9), a drugi jest dodatkowo wprowadzoną modyfikacją:

$$w_q = \begin{cases} \dfrac{d_{max} - d_q}{d_{max} - d_{min}} \cdot \dfrac{d_{max} + d_q}{d_{max} + d_{min}} & \text{dla } d_{max} \neq d_{min} \\ 1 & \text{dla } d_{max} = d_{min} \end{cases} . \tag{6.10}$$

Metoda EWk-NN

Kolejna z modyfikacji, nazwana EWk-NN (*ang.* Exponential Weighted k-NN), przedstawiona została w pracy [30] i polega na wprowadzeniu członu wykładniczego do formuły (6.10). Zdaniem jej autorów, poprawia ona jakość klasyfikacji w przypadku danych referencyjnych z nie-

równoliczną reprezentacją klas (klasy nie są zrównoważone).

$$w_q = \begin{cases} \exp(-(\frac{d_{max} - d_q}{d_{max} - d_{min}} \cdot \frac{d_{max} + d_q}{d_{max} + d_{min}})) & \text{dla } d_{max} \neq d_{min} \\ 1 & \text{dla } d_{max} = d_{min} \end{cases} . \quad (6.11)$$

6.2 Losowe próbkowanie przestrzeni cech (RSk-NN)

Metoda losowych podzbiorów przestrzeni cech (*RS – ang.* Random Subspace) wykorzystuje strategię łączenia decyzji prostych klasyfikatorów, co pozwala na uzyskanie lepszej globalnej jakości klasyfikacji, niemożliwej do uzyskiwana przez dowolny pojedynczy klasyfikator z puli. Tego typu połączenie klasyfikatorów nazywane jest komitetem klasyfikacyjnym, zespołem klasyfikatorów lub systemem wieloklasyfikatorowym [33, 67, 37, 12, 59].

Niech zbiór danych wejściowych X składa się z Q obiektów, z których każdy opisany jest wektorem F cech: $\mathbf{x}_q = [x_q^{(1)}, \ldots, x_q^{(F)}]$, $\mathbf{x}_q \in X$, $q \in \{1, \ldots, Q\}$. Wybierzmy losowo $r < F$ cech z każdego wektora $\mathbf{x}_q$. Uzyskany w ten sposób Q elementowy zbiór $\tilde{X}^r$ zawiera r-wymiarowe wektory $\tilde{\mathbf{x}}_q^r = [x_q^{(1)}, \ldots, x_q^{(r)}]$. Poszczególne elementy wektorów $\tilde{\mathbf{x}}_q^r$ są losowo wybranymi elementami $x_q^{(f)}, f \in \{1, \ldots, F\}$ wektorów $\mathbf{x}_q$. Tworzymy $M \in \mathbb{N}^+$ takich zbiorów, których używamy do wytrenowania M klasyfikatorów. Decyzja komitetu wyłaniana jest na podstawie głosowania większościowego decyzji klasyfikatorów wchodzących w skład komitetu. Dla powyższych założeń, algorytm klasyfikacji RS przedstawić można w postaci pseudokodu:

Algorytm 5: Algorytm klasyfikacji bazujący na losowych podprzestrzeniach cech

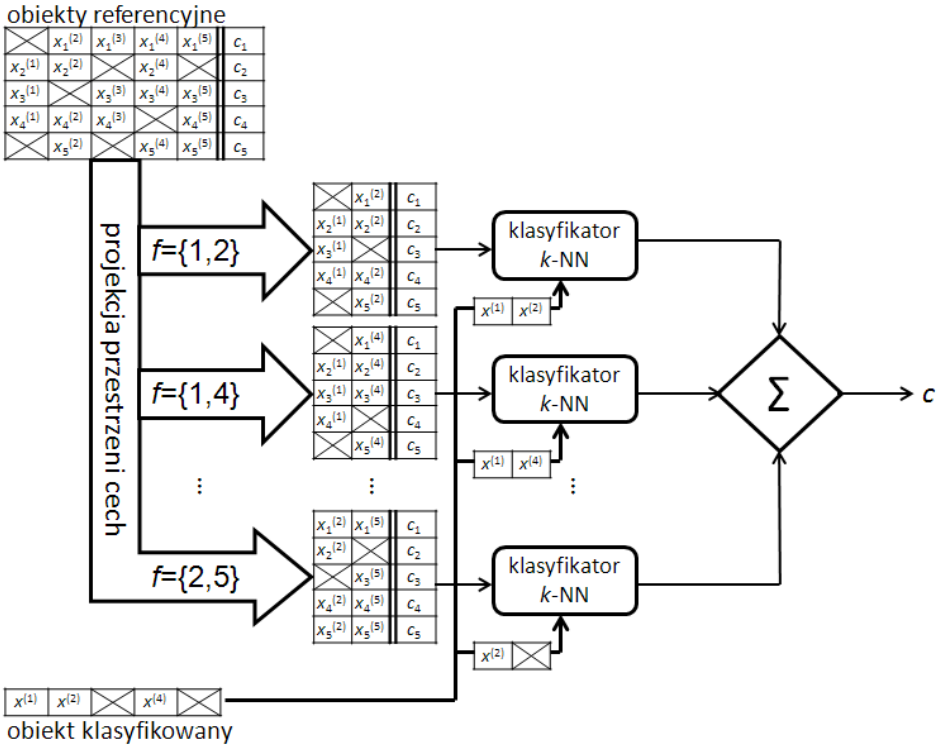
```
1 begin
2     for m := 1 to M do
3         Wybierz losowo r cech z przestrzeni X^F, tworząc nową przestrzeń X̃^r;
4         Stwórz klasyfikator D^m(x) wytrenowany na elementach (cechach) przestrzeni X̃^r;
5     end
6     Wyznacz decyzję komitetu, na podstawie głosowania większościowego decyzji
      klasyfikatorów składowych: D(x) = argmax_{c∈C} ( Σ_{m=1}^{M} (I(D^m(x) = c)) );
7     return D(x);
8 end
```

Podstawową zaletą losowego tworzenia podzbioru cech jest jej elastyczność. Może być użyta w połączeniu z różnymi rodzajami klasyfikatorów, a w szczególności z leniwymi klasy-

fikatorami minimalno-odległościowymi k-NN [26], co będzie pokazane poniżej. Ten typ klasyfikacji oznaczany będzie akronimem RSk-NN. Do wad metody zaliczyć można zwiększone zapotrzebowanie na pamięć i moc obliczeniową, gdyż komitet składa się z wielu klasyfikatorów działających w oparciu o losowe wybrane podzbiory cech, pierwotnego zbioru uczącego. Łączny wymiar wszystkich podprzestrzeni cech może znacząco przewyższać wymiar pierwotnego zbioru uczącego, gdyż cechy są losowo wybierane ze zwracaniem.

Jeśli przyjąć że pierwotny zbiór uczący opisany jest 10 cechami, a komitet będzie się składał z 10 klasyfikatorów, z których każdy będzie działał w oparciu o losowy wybór 5 cech, to komitet będzie operował na 50 cechach, z których część może się powtarzać w poszczególnych klasyfikatorach.

Przykład budowy i sposobu wyłaniania decyzji komitetu RSk-NN dla zbioru referencyjnego zbudowanego z $Q = 5$ obiektów opisanych $F = 5$ cechami, w oparciu o które wygenerowano M podzbiorów treningowych o $r = 2$ cechach, pracującego z brakującymi wartościami zarówno w zbiorze treningowym jak i testowym przedstawiono na Rys. 6.1.



Rys. 6.1: Schemat budowy proponowanego komitetu klasyfikatorów RSk-NN (przekreślone pole oznacza niezdefiniowaną wartość cechy).

Oszacowanie liczby zdegradowanych wektorów uczących dla klasyfikatora RSk-NN działającego na niepełnych danych uczących

Ponieważ wektory $\tilde{\mathbf{x}}_q^r$ są tworzone losowo, mogą zawierać różne zestawy cech – z konkretnymi wartościami, jak i wartościami *null*. Uczenie klasyfikatora przebiega tym lepiej, im mniej wekto-

rów danych zawierać będzie wartości *null*. *Null*, jako wartość nieokreślona (pusta), oznacza brak informacji o danej cesze, co powoduje, że klasyfikator nie będzie prawidłowo uczony i popełniać będzie błędy większe, niż w przypadku gdyby był uczony na komplecie cech.
Błąd ten można oszacować. Niech liczba elementów wektora $\mathbf{x}_q$ wynosi F, a liczba elementów $\neq$ *null* wynosi n. Wtedy $F - n$ jest liczbą elementów $=$ *null* w tym wektorze.
Wprowadźmy oznaczenia:
F – liczba elementów (cech) pierwotnego wektora danych $\mathbf{x}_q$,
n – liczba elementów $\neq$ *null* w wektorze $\mathbf{x}_q$,
r – liczba losowo wybranych cech z wektora $\mathbf{x}_q$, tworzących wektor $\tilde{\mathbf{x}}_q^r$,
Niech N oznacza liczbę wartości $\neq$ *null* w wektorze $\tilde{\mathbf{x}}_q^r$, wtedy prawdopodobieństwo pojawienia się takiej wartości wyznaczyć można na podstawie następującego wzoru:

$$P(N = i) = \frac{\binom{n}{i} \cdot \binom{F-n}{r-i}}{\binom{F}{r}}, \qquad i = 0, \ldots, r. \tag{6.12}$$

Prawdopodobieństwo, że wektor $\tilde{\mathbf{x}}_q^r$ składać się będzie wyłącznie z wartości $=$ *null* wyniesie:

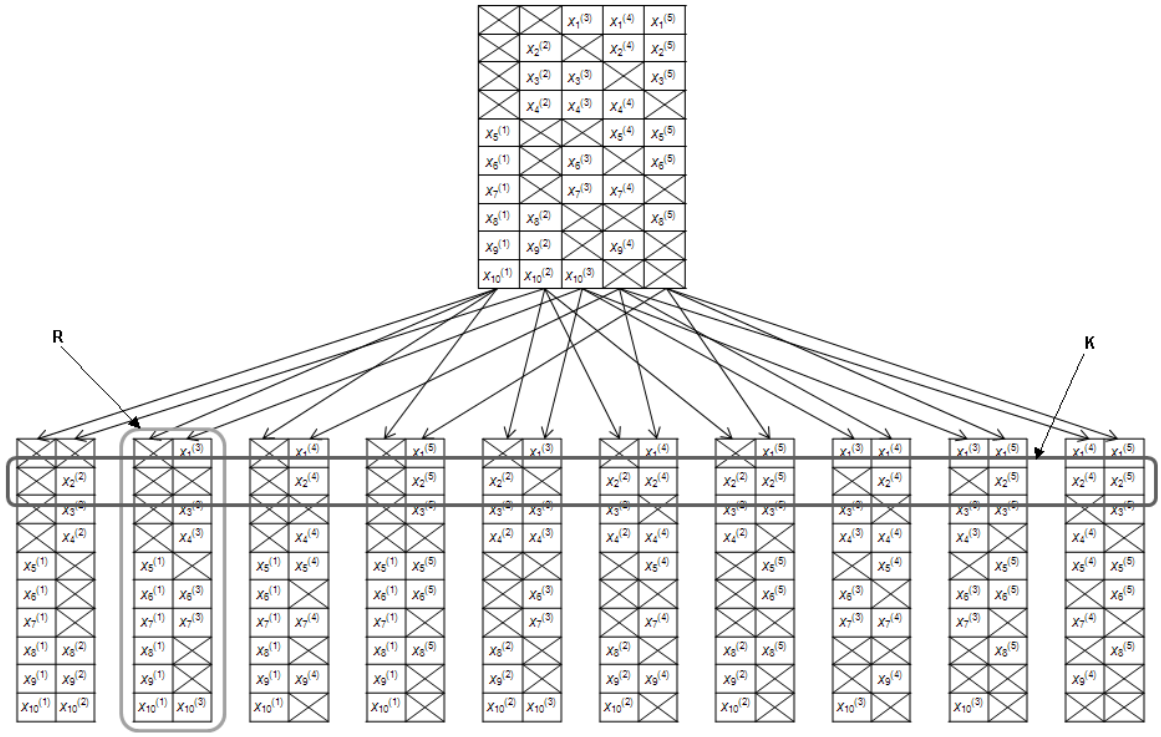
$$P(N = 0) = \frac{\binom{F-n}{r}}{\binom{F}{r}}. \tag{6.13}$$

Niech U będzie prawdopodobieństwem, że w dowolnym wektorze $\tilde{\mathbf{x}}_q^r$ znajdzie się wartość $\neq$ *null*. Wtedy:

$$P(U) = 1 - P(N = 0). \tag{6.14}$$

Przedstawione powyżej wzory na prawdopodobieństwo wystąpienia określonych zdarzeń objaśnić można za pomocą przykładu.

Przykład rozmieszczenia wartości cech w klasyfikatorze RSk-NN
Niech liczba cech pierwotnych danych wynosi $F = 5$. Liczba cech $\neq$ *null* wynosi $n = 3$, liczba cech $=$ *null* wynosi $F - n = 2$. Dokonujemy losowego wyboru $r = 2$ cech z pośród F cech. Pierwotny rozkład cech jest nieznany więc można wylosować $\binom{F}{r}$ układów dwóch cech. Niech liczba klasyfikatorów bazowych tworzących komitet wynosi $M = 10$. Dla takich założeń, rozmieszczenie wartości poszczególnych cech przedstawia Rys. 6.2. Na rysunku zaznaczono rozmieszczenie wartości *null* (przekreślone pola) w wektorach klasyfikowanych ($\mathbf{K}$) przykładowego obiektu, oraz w zbiorze referencyjnym ($\mathbf{R}$) jednego z klasyfikatorów wchodzących w skład komitetu. Prawdopodobieństwa wynoszą odpowiednio: $P(N = 0) = 0{,}1$, $P(N = 1) = 0{,}6$, $P(N = 2) = 0{,}3$, $P(U) = 0{,}9$.



Rys. 6.2: Przykład rozkładu cech pomiędzy klasyfikatorami komitetu RSk-NN.

Rozdział 7

Opis proponowanych metod klasyfikacji

Proponowane w Rozprawie metody klasyfikacji danych wielowymiarowych dotyczą przede wszystkim przypadków, kiedy dane wejściowe są niekompletne — z brakującymi wartościami cech i niezrównoważoną licznością klas. Brakujące dane mogą występować zarówno w wektorach referencyjnych jak i klasyfikowanych, a ich charakter jest całkowicie losowy.

7.1 Wstępne usuwanie brakujących cech i niepełnych wektorów (AFFk-NN)

Pierwsza z proponowanych modyfikacji klasyfikatora k-NN na potrzeby klasyfikacji niepełnych danych opiera się o wstępne filtrowanie danych. Jest to połączenie eliminacji cech nieposiadających przypisanych wartości oraz usuwania niepełnych wektorów. W proponowanej metodzie wstępnej filtracji danych niepełnych można wyróżnić trzy etapy eliminacji niepełnych danych:

Etap 1: Z wektora klasyfikowanego $\mathbf{x}$ usuwane są cechy, które posiadają wartości *null* ($x^{(f)} = null$), tworząc wektor $\tilde{\mathbf{x}}$. Niech F oznacza pierwotną liczbę cech, zaś u oznacza liczbę usuniętych cech.

Etap 2: Ze wszystkich Q wektorów referencyjnych $\mathbf{x}_q$ usuwane są cechy które nie występują w zredukowanym wektorze klasyfikowanym $\tilde{\mathbf{x}}$.

Etap 3: Zbiór referencyjny poddawany jest operacji filtrowania niepełnych wektorów - usuwane są wektory, w których brakuje co najmniej jednej, spośród pozostałych $(F-u)$, wartości cechy ($x_q^f = null$).

Tak zredukowany wektor klasyfikowany $\tilde{\mathbf{x}}$ podawany jest na wejście klasyfikatora k-NN, trenowanego na $(F-u)$ wymiarowym zbiorze uczącym, pozbawionym niepełnych wektorów.
Metoda ta nie ingeruje w strukturę klasyfikatora, zatem może być stosowana z dowolnym klasyfikatorem. Ze względu na filtrowanie przestrzeni cech zbioru uczącego w zależności od cech

występujących w wektorze klasyfikowanym jest szczególnie przydatna w połączeniu z klasyfikatorami leniwymi. W dalszej części Rozprawy nazywana będzie AFFk-NN (od *ang.* Adaptive Feature Filtering).

7.2 Klasyfikacja w oparciu o oddzielne cechy (SFk-NN)

Metoda ta łączy cechy metody AFFk-NN oraz metody losowych podprzestrzeni cech RSk-NN. Proponowany klasyfikator jest klasyfikatorem leniwym, czyli przechowującym wszystkie obiekty referencyjne (uczące), względem których wyznacza odległość do obiektu klasyfikowanego.
Konstrukcja klasyfikatora pracującego w środowisku danych niepełnych, w których przedziały zmienności są często nierozłączne, a liczność klas niezrównoważona, jest zadaniem trudnym. W niniejszej pracy zaproponowano rozwiązanie, które umożliwia budowę klasyfikatora pracującego w oparciu o takie dane. Proponowane rozwiązanie bazuje na jednocechowych wektorach referencyjnych.

Przyjmijmy, że zbiór danych uczących $\mathbf{R}$, opisany jest następującą macierzą o rozmiarach $Q \times (F+1)$:

$$\mathbf{R} = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(F)} & c_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_Q^{(1)} & x_Q^{(2)} & \cdots & x_Q^{(F)} & c_Q \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 & c_1 \\ \vdots & \vdots \\ \mathbf{x}_Q & c_Q \end{bmatrix}. \tag{7.1}$$

W procesie próbkowania przestrzeni cech możliwe jest stworzenie F zbiorów referencyjnych $R^{(f)}$ zawierających wartość f-tej cechy i skojarzoną z nią etykietę klasy:

$$\mathbf{R}^{(f)} = \begin{bmatrix} x_1^{(f)} & c_1 \\ \vdots & \vdots \\ x_Q^{(f)} & c_Q \end{bmatrix}, \quad f = 1, \ldots, F. \tag{7.2}$$

Opisywana projekcja cech eliminuje problem brakujących wartości cech, zarówno w wektorze danych klasyfikowanych, jak i w wektorach danych uczących. Wartości cech wektora klasyfikowanego są odrębnie porównywane z wartościami odpowiadających im cech ze zbioru wektorów referencyjnych. Uwzględniane są tylko te wektory referencyjne, w których analizowana cecha posiada określoną wartość (różną od *null*). W efekcie można zbudować komitet równolegle działających klasyfikatorów k-NN operujących na rozłącznych, jednowymiarowych podzbiorach przestrzeni cech klasyfikowanego obiektu. Komitet ten nazywany będzie klasyfikatorem SFk-NN (*ang.* Separate Features k-NN).
Działanie komitetu przebiega zgodnie z następującymi etapami:

Etap 1: Z wektora klasyfikowanego $\mathbf{x}$ oraz ze wszystkich F-elementowych wektorów referencyjnych

$\{\mathbf{x}_q\}_{q=1}^{Q}$ usuwane są cechy, które w wektorze klasyfikowanym posiadają wartości *null* (tzn. $x^{(f)} = null$). Niech u oznacza liczbę usuniętych cech.

Etap 2: Tworzony jest komitet złożony z $(F-u)$ klasyfikatorów. Dla każdego z nich tworzony jest zbiór obiektów referencyjnych. Zbiór taki liczy Q obiektów, czyli tyle ile pierwotny zbiór referencyjny. Obiekty te są jednoelementowymi wektorami zawierającymi wartość f-tej cechy z pierwotnych wektorów referencyjnych ($\mathbf{x}_q^f = [x_q^{(f)}], q = (1, \ldots, Q)$).

Etap 3: Z każdego z $(F-u)$ zbiorów referencyjnych usuwane są jednoelementowe wektory, które nie zawierają zarejestrowanej wartości cechy (tzn. $\mathbf{x}_q^f = [null]$).

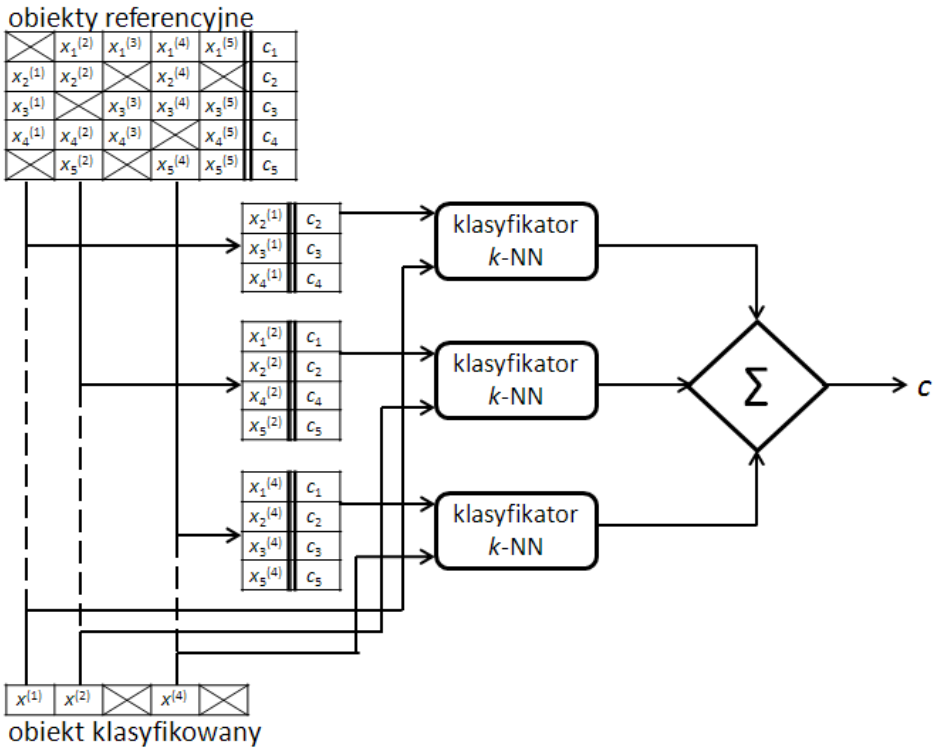
Etap 4: Każda z pozostałych $(F-u)$ cech wektora klasyfikowanego poddawana jest klasyfikacji przez odpowiadający jej klasyfikator wchodzący w skład komitetu.

Etap 5: Każdy z $(F-u)$ klasyfikatorów zwraca wartości wsparcia dla każdej z L klas. Opcjonalnie wartości te mogą być mnożone przez współczynniki wagowe wyznaczone dla poszczególnych cech i klas.

Etap 6: Wartości wsparcia dla każdej z L klas są sumowane i wybierana jest klasa c, dla której suma wartości wsparcia jest największa. Klasa ta jest zwracana jako decyzja komitetu klasyfikatorów.

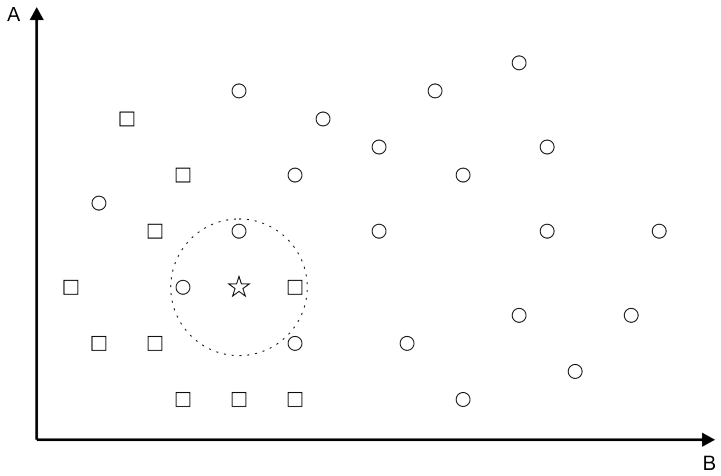
Ze względów implementacyjnych Etapy 1) oraz 3) są możliwe do pominięcia. Jeśli przyjmiemy, że klasyfikator odpowiadający cesze f, której wartość nie została zarejestrowana w wektorze klasyfikowanym, nie zwróci żadnej decyzji lub zwrócone przez niego wartości wsparcia v_c^f dla poszczególnych L klas będą równe 0 (tzn. $[v_c^f]_{c=1}^{L} = 0$), możemy pominąć Etap 1 i zawsze tworzyć komitet składający się z F klasyfikatorów. Również wektory referencyjne nie zawierające żadnej wartości nie zostaną wybrane jako najbliżsi sąsiedzi obiektu klasyfikowanego $\mathbf{x}$, można zatem pominąć Etap 3.
Warto zauważyć, że sortując zbiory referencyjne poszczególnych klasyfikatorów zgodnie z wartościami cechy $x_q^{(f)}$ znacząco upraszcza się proces wyszukiwania najbliższych sąsiadów. Nie ma potrzeby każdorazowego sortowania tablicy odległości, wystarczy zrobić to raz, w momencie tworzenia zbioru referencyjnego. Wtedy, szukając najbliższych sąsiadów projekcji f-tej cechy obiektu klasyfikowanego $\mathbf{x}$ wystarczy znaleźć najbliższego sąsiada, a następnie porównać z k elementami wstecz i w przód.
Kolejną zaletą opisywanego klasyfikatora SFk-NN jest możliwość równoległego wykonywania algorytmu (dla każdej z cech) co może dodatkowo skrócić czas obliczeń.

Przykład budowy i wyłaniania decyzji takiego komitetu dla obiektów opisanych 5-cioma cechami, z brakującymi wartościami zarówno w zbiorze treningowym jak i testowym przedstawiono na Rys. 7.1.



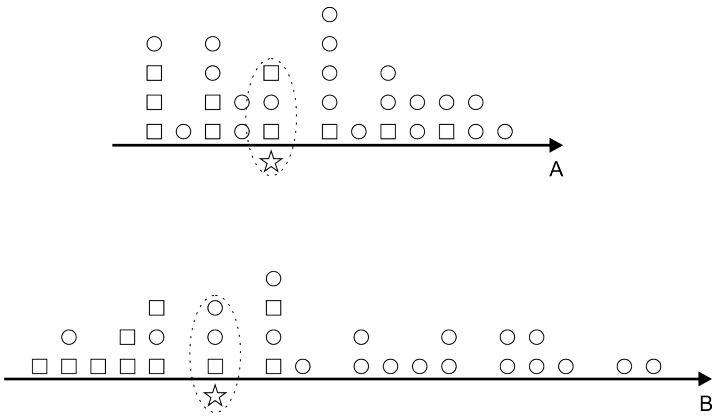
Rys. 7.1: Schemat budowy proponowanego komitetu klasyfikatorów SFk-NN.

Wartości cech porównywane są zgodnie z działaniem klasycznego klasyfikatora k-NN, jednak z uwagi na podział przestrzeni cech i proces kwantyzacji danych, należało uwzględnić możliwość wystąpienia jednakowych odległości wartości referencyjnych od wartości klasyfikowanej. Na Rys. 7.2 zilustrowano zasadę działania algorytmu SFk-NN (dla k=3), analizującego dwuwymiarowe wektory danych. Niech klasyfikowany obiekt $\mathbf{x}$ opisany jest wektorem dwóch cech $\mathbf{x} = [A, B]$. Zatem obiekt $\mathbf{x}$ oznaczony jako „☆”, zostanie przypisany do klasy oznaczonej jako „○” , gdyż dwa najbliżej leżące od niego obiekty należą do klasy „○”, a jeden do klasy „□”.



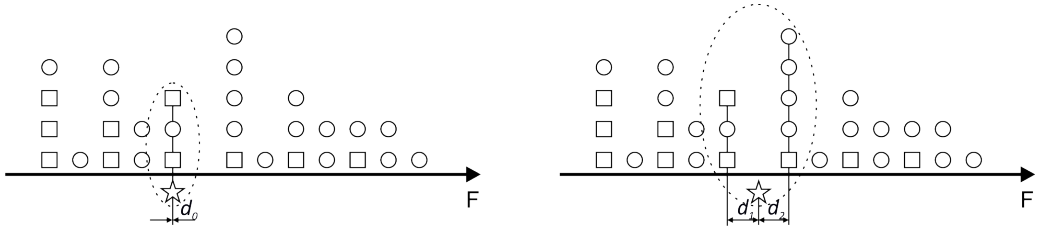
Rys. 7.2: Graficzna reprezentacja standardowego algorytmu wyboru najbliższych sąsiadów.

Kolejny rysunek (Rys. 7.3) pokazuje oddzielne rzutowanie cechy A na oś przestrzeni A, oraz cechy B na oś przestrzeni B. Każda z osi opisuje jeden wymiar przestrzeni cech. Można zauważyć, że po rzutowaniu cech na osie A i B klasa, do której należy klasyfikowany obiekt zostanie w przestrzeni A określona jako „□", natomiast w przestrzeni B jako „○". Oznacza to, że wynik klasyfikacji jest nierozstrzygnięty. Tę niejednoznaczność można zlikwidować dzięki opisanej w Podrozdziale 7.3 modyfikacji polegającej na przyporządkowaniu obiektom referencyjnym należącym do poszczególnych klas, różnych wag.



Rys. 7.3: Graficzna reprezentacja rzutowania wartości cech i wyboru najbliższych sąsiadów w przestrzeni jednowymiarowej.

W proponowanym w Rozprawie algorytmie SFk-NN liczba rozpatrywanych najbliższych sąsiadów jest zmienna, lecz nie mniejsza niż k. W przypadku, gdy oprócz k najbliższych obiektów istnieją inne obiekty, jednakowo blisko odległe od klasyfikowanego obiektu, w procesie klasyfikacji uwzględnione zostaną wszystkie te obiekty. Rysunek 7.4 przedstawia przykład dla $k = 1$, gdzie po lewej uwzględniono 3 najbliższe ($d_0 = 0$) do klasyfikowanego obiektu „☆" obiekty referencyjne, natomiast po prawej uwzględniono 8 najbliższych ($d_1 = d_2$) do obiektu „☆" obiekty referencyjne. Jako miarę odległości d przyjęto tutaj odległość euklidesową.



Rys. 7.4: Ilustracja zasady zwiększania liczby najbliższych sąsiadów w klasyfikatorze 1-NN.

Dla danych wielowymiarowych, reprezentowanych macierzą (7.1) budowany jest komitet składający się z F klasyfikatorów k-NN. Klasyfikatory pracują niezależnie, każdy w oparciu o inną cechę $x^{(f)}$ obiektu $\mathbf{x}$, $f \in \{1, 2, \ldots, F\}$.

Jeśli przyjąć, że liczba uwzględnianych sąsiadów jest wyrażona jako:

$$Q^f = card\left\{\bar{\mathbf{R}}^{(f)}\right\}, \tag{7.3}$$

gdzie:
$\bar{\mathbf{R}}^{(f)}$ – zbiór najbliższych do wektora $\mathbf{x}$ sąsiadów, wyznaczony z uwzględnieniem cechy f.
Dla klasyfikowanego wektora $\mathbf{x}$, w którym rozpatrywana jest tylko cecha f, można obliczyć wartość wsparcia $v_c^f(\mathbf{x})$ dla klasy c, według następującej formuły:

$$v_c^f(\mathbf{x}) = \frac{card\left\{\bar{\mathbf{x}}_q^{(f)} : \bar{c}_q = c\right\}_{q=1}^{Q^f}}{Q^f}, \tag{7.4}$$

gdzie:
$\mathbf{x}$ – klasyfikowany wektor cech,
$\bar{\mathbf{x}}_q^{(f)}$ – wektor ze zbioru najbliższych sąsiadów klasyfikowanego obiektu $\mathbf{x}$,
c – identyfikator klasy, $c \in \{1, \ldots, L\}$,
$\bar{c}_q$ – identyfikator klasy obiektu $\bar{\mathbf{x}}_q$ ze zbioru $\bar{\mathbf{R}}^{(f)}$,
f – rozpatrywana cecha $x^{(f)} \in \mathbf{x}$.

W celu wyłonienia globalnej wartości funkcji wsparcia $V_c(\mathbf{x})$ klasyfikatora komitetowego $D(\mathbf{x})$, wartości funkcji wsparcia $v_c^f(\mathbf{x})$ klasyfikatorów składowych są uśredniane:

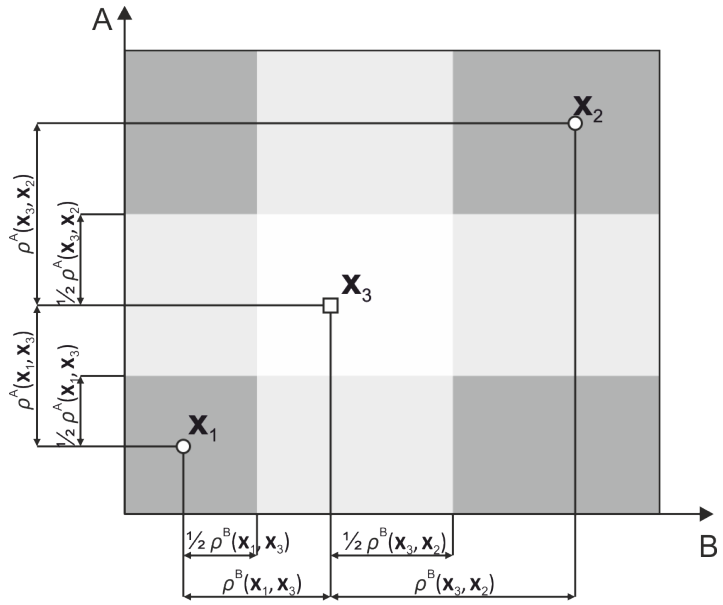
$$V_c(\mathbf{x}) = \frac{1}{F}\sum_{f=1}^{F} v_c^f(\mathbf{x}). \tag{7.5}$$

Klasyfikacja wektora $\mathbf{x}$ polega na określeniu etykiety klasy, dla której globalna wartość funkcji wsparcia klasyfikatora komitetowego $D(\mathbf{x})$, składającego się z F odrębnych klasyfikatorów jest największa:

$$D(\mathbf{x}) = \underset{c\in\{1,\ldots,L\}}{\operatorname{argmax}}\left(V_c(\mathbf{x})\right). \tag{7.6}$$

Ten sposób wyłaniania decyzji komitetu klasyfikatorów nosi nazwę uśredniania Bayesowskiego [27].

Przy takiej konstrukcji klasyfikatora możliwe jest występowanie remisów w fazie sumowania decyzji. Przykład takiej sytuacji ilustruje rysunek (Rys. 7.5). Występują na nim trzy obiekty referencyjne, dwa ($\mathbf{x}_1$ i $\mathbf{x}_2$) należące do klasy „◯", oraz jeden ($\mathbf{x}_3$) należący do klasy „□". Niech $k = 1$. Dla takich założeń, jeśli klasyfikowany obiekt znajdzie się w obszarze białym, zostanie sklasyfikowany jako obiekt należący do klasy „□". Jeśli natomiast znajdzie się w obszarze ciemnoszarym, zostanie sklasyfikowany, jako należący do klasy „◯". Obiekty trafiające do obszaru jasnoszarego nie zostaną sklasyfikowane jednoznacznie. W celu rozwiązania remisu obiekty zostaną przypisane do losowej klasy.



Rys. 7.5: Przykład komitetu klasyfikatorów 1-NN działających w oparciu o wartości cechy A oraz B („○" i „□" - obiekty referencyjne, obszar ciemnoszary - sklasyfikowany jako klasa „○", obszar jasnoszary - brak jednoznacznego wyniku, obszar biały - sklasyfikowany jako klasa „□").

7.3 Modyfikacja bazująca na współczynniku wagowym (SFk-NN/C)

W przypadku, gdy liczba obiektów referencyjnych poszczególnych klas nie jest jednakowa omawiany klasyfikator będzie wykazywał tendencję do zaliczania obiektów do najliczniej reprezentowanej klasy w zbiorze danych referencyjnych. Aby temu zapobiec, wprowadzono dodatkowy współczynnik równoważenia klas w postaci wagi w_c, która uwzględnia liczność poszczególnych klas w zbiorze danych referencyjnych.
Niech zbiór danych treningowych oznaczony będzie jako $\mathbf{R}$, wtedy opisaną powyżej niekorzystną tendencję działania klasyfikatora niwelować można za pomocą współczynnika wagi:

$$w_c = 1 - P(c|\mathbf{x}_q) = 1 - \frac{card\left\{\mathbf{x}_q \in \mathbf{R} : c_q = c\right\}}{card\left\{\mathbf{R}\right\}}, \tag{7.7}$$

gdzie $P(c|\mathbf{x}_q)$ jest prawdopodobieństwem przynależności obiektu $\mathbf{x}_q$, ze zbioru referencyjnego $\mathbf{R}$, do klasy c.

Współczynnik w_c można interpretować jako prawdopodobieństwo zdarzenia przeciwnego do wylosowania wektora należącego do danej klasy spośród wszystkich wektorów referencyjnych. Można wyznaczyć ten współczynnik globalnie dla całego zbioru uczącego, lub lokalnie dla zbioru uczącego każdego z F klasyfikatorów wchodzących w skład komitetu (wtedy, oznaczany będzie w_c^f). Zatem funkcja wsparcia klasyfikatora komitetowego również może zostać obliczona na dwa

sposoby, uwzględniając współczynnik wagi wyznaczany globalnie lub lokalnie:

a) ***globalnie*** – dla wszystkich obiektów zbioru uczącego:

$$V_c(\mathbf{x}) = w_c \cdot \frac{1}{F} \sum_{f=1}^{F} v_c^f(\mathbf{x}), \tag{7.8}$$

b) ***lokalnie*** – dla poszczególnych cech obiektu zbioru uczącego:

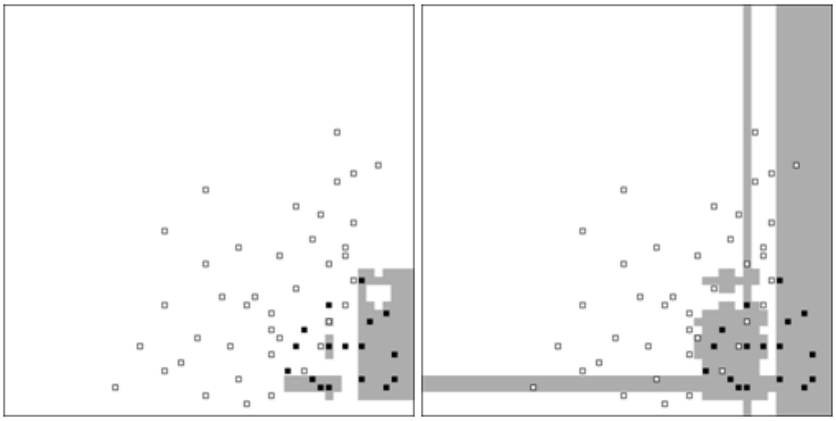
$$V_c(\mathbf{x}) = \frac{1}{F} \sum_{f=1}^{F} \left(w_c^f \cdot v_c^f(\mathbf{x}) \right). \tag{7.9}$$

Tak samo jak w metodzie SFk-NN klasyfikacja wektora $\mathbf{x}$ polega na określeniu etykiety klasy, dla której globalna wartość funkcji wsparcia klasyfikatora komitetowego $D(\mathbf{x})$, składającego się z m odrębnych klasyfikatorów jest największa:

$$D(\mathbf{x}) = \underset{c \in \{1,\ldots,L\}}{\operatorname{argmax}} \left(V_c(\mathbf{x}) \right). \tag{7.10}$$

Stanowi to realizację czwartego celu pracy.

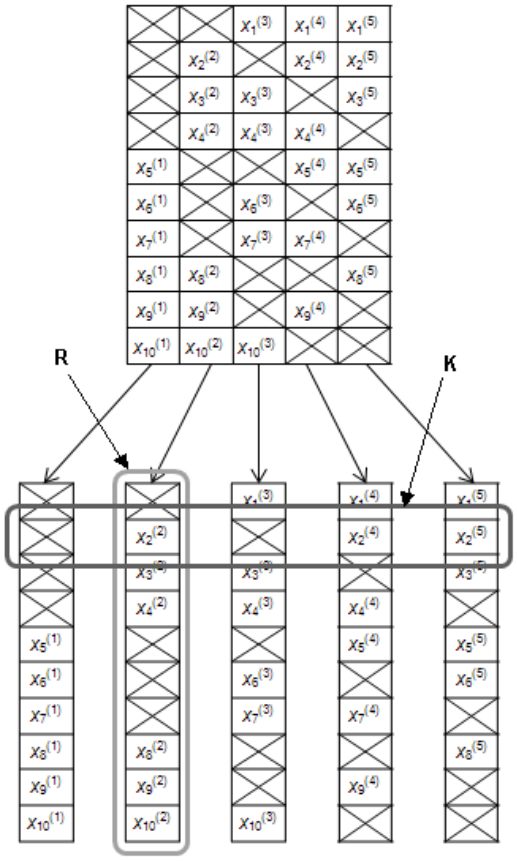
Na rysunku 7.6 pokazano efekt działania zmodyfikowanego komitetowego klasyfikatora SFk-NN/C. Na ilustracjach widoczne są zbiory treningowe składające się z obiektów należących do dwóch klas: klasy większościowej „□” (40 obiektów) oraz klasy mniejszościowej „■” (17 obiektów). Obszary białe zostały sklasyfikowane, jako należące do klasy „□”, natomiast obszary szare, jako należące do klasy „■”. Swoistość klasyfikacji dla klasy mniejszościowej bez wprowadzenia systemu wag wynosiła 70%, natomiast zmodyfikowana metoda z systemem wag pozwoliła uzyskać 100% swoistości, pomimo że klasyfikator nie został zaprojektowany do klasyfikowania danych o tak małej liczbie wymiarów. Oznacza to, że wprowadzone modyfikacje poprawiły zdolność klasyfikatora do poprawnej oceny obiektów klas mniejszościowych.



Rys. 7.6: Porównanie działania klasyfikatora SFk-NN (po lewej) oraz jego odmiany z wagami dla klas SFk-NN/C (po prawej). Klasy: biała, czarna; obszar biały – sklasyfikowany jako klasa „biała”, obszar szary – sklasyfikowany jako klasa „czarna”.

7.4 Przykład rozmieszczenia wartości cech w klasyfikatorze SFk-NN(/C)

Podobnie jak w przypadku klasyfikatora RSk-NN możliwe jest oszacowanie prawdopodobieństwa wystąpienia wartości *null* w wektorach klasyfikowanych oraz w zbiorach referencyjnych poszczególnych klasyfikatorów wchodzących w skład klasyfikatora komitetowego SFk-NN(/C). Do obliczeń wykorzystujemy wzory 6.12, 6.13 oraz 6.14 wyprowadzone już wcześniej.
Niech liczba cech pierwotnych danych wynosi $F = 5$. Liczba cech $\neq$ *null* wynosi $n = 3$, liczba cech $=$ *null* wynosi $F - n = 2$. Dokonujemy losowego wyboru $r = 1$ cech z pośród F cech. Pierwotny rozkład cech jest nieznany więc można wylosować $\binom{F}{r}$ układów jednej cechy.
Rozmieszczenie wartości poszczególnych cech przedstawia Rys. 7.7. Na rysunku zaznaczono rozmieszczenie wartości *null* w wektorach klasyfikowanych ($\mathbf{K}$) przykładowego obiektu, oraz w zbiorze danych referencyjnych ($\mathbf{R}$) jednego z klasyfikatorów wchodzących w skład komitetu. Niech N oznacza liczbę wartości $\neq$ *null* w wektorze $\tilde{\mathbf{x}}_q^r$, wtedy prawdopodobieństwa wynoszą odpowiednio: $P(N = 0) = 0{,}4$, $P(N = 1) = 0{,}6$, $1 - P(N = 0) = 0{,}6$.



Rys. 7.7: Przykład rozkładu danych w bazowych homogenicznych klasyfikatorach k-NN komitetu SFk-NN.

7.5 Selekcja cech w oparciu o klasyfikator SFk-NN/C

Budowa przedstawionego w niniejszym Rozdziale klasyfikatora SFk-NN/C sprawia, że w oparciu o niego można stworzyć metodę selekcji cech. Proponowana metoda selekcji cech opiera się na rankingu cech wyznaczonym w procesie krzyżowej walidacji zbioru referencyjnego w oparciu o pojedyncze cechy f i jedną z miar oceny jakości klasyfikacji (Tabela 3.2). Zgodnie z systematyką przedstawioną w Rozdziale 5 jest to zarazem metoda rankingowa oraz opakowana.
Tworząc ranking cech w oparciu o wybrane miary jakości klasyfikacji, można przeprowadzić selekcję cech na dwa sposoby:

a) Ustalając wartość progową wybranej miary jakości klasyfikacji, poniżej której cecha nie będzie brana pod uwagę w procesie klasyfikacji obiektów.
b) Wybierając metodą rankingu liczbę cech o największej wartości miary jakości klasyfikacji.

Na potrzeby eksperymentu użyty zostanie wariant b), a optymalna liczba cech zostanie dobrana metodą *brute-force*. W ramach badań sprawdzone będą dwie metody tworzenia rankingu cech:

a) Pierwsza z nich będzie procedurą indywidualnego rankingu cech opisaną Algorytmem 4 w Rozdziale 5.
b) Druga będzie rankingiem cech dla każdej klasy.

W celu utworzenia odrębnego rankingu cech dystynktywnych dla każdej z L klas, zbiór danych zostanie poddany walidacji krzyżowej typu *leave one out*. Z uwagi na możliwe niezrównoważenie liczby obiektów w poszczególnych klasach, zamiast typowo wykorzystywanej do budowy list rankingowych miary *ogólnej dokładności*, dla każdej z klas zostaną wyznaczone współczynniki $F - Measure$ i $G - Measure$. Drogą eksperymentalną zostanie wybrany jeden z nich. W oparciu o wartości wybranego współczynnika stworzonych zostanie L list rankingowych cech. Jest to metoda podobna do modyfikacji etykiet klas w zbiorach danych mająca na celu zastąpienie klasyfikacji L klasowej, przez L odrębnych klasyfikacji binarnych. W przeciwieństwie do modyfikacji etykiet klas, użyta metoda wymaga jedynie jednokrotnego wykonania walidacji krzyżowej zbioru danych.
Poszczególne listy rankingowe będą charakteryzowały zdolność klasyfikatora do odróżniania obiektów należących do jednej z L klas, od obiektów należących do pozostałych klas. Z każdej z L tak powstałych list rankingowych wybrana będzie jednakowa liczba najlepszych cech. Cechy wybrane z jednej listy rankingowej będą usuwane z pozostałych list. Zatem, z każdej z L list rankingowych, zostanie wybrana jednakowa liczba niepowtarzających się cech.

Rozdział 8

Badania eksperymentalne

W niniejszej pracy opisano wyniki serii eksperymentów na sztucznie zmodyfikowanych danych pochodzących z rzeczywistych obserwacji. Modyfikacje danych miały na celu odzwierciedlenie niedoskonałości występujących w rzeczywistych danych, które mogą zaburzać proces klasyfikacji. Najlepszym przykładem takich problematycznych danych mogą być dane medyczne. Rozważania można jednak przenieść na dowolny inny rodzaj danych, w których występuje niezrównoważenie klas lub występują braki danych.
Dane, na których uczony będzie klasyfikator mogą pochodzić z niepełnych źródeł historycznych lub być pełne. Dane, które mają być klasyfikowane w procesie testowania również mogą być pełne lub niekompletne. Występuje więc wielowariantowość scenariuszy badania jakości klasyfikacji danych. W pracy przyjęto trzy scenariusze, według których można uczyć i testować klasyfikator – będą one przedstawione w dalszej części tego Rozdziału.
Testy przeprowadzono w oparciu o zestawy wektorów o różnej liczbie brakujących cech $x^{(f)}$ występujących w wektorach cech $\mathbf{x}$. Przez brakujące cechy, należy rozumieć cechy, których wartość nie została zarejestrowana i mają przypisaną wartość *null*.
W przeprowadzonych testach porównawczych analizowane były różne odmiany klasyfikatorów k-NN, w tym klasyfikatory komitetowe RSk-NN i opracowany na potrzeby niniejszej Rozprawy klasyfikator komitetowy SFk-NN(/C). Na potrzeby oceny jakości klasyfikacji wykorzystane zostały bazy danych o następujących charakterystykach:

Tabela 8.1: Charakterystyka baz danych użytych w eksperymentach.

L.p.	Pełna nazwa	Skrót	Obiekty (Q)	Cechy (F)	Klasy (L)
1.	Wine	WINE	178 (144)*	13	3
2.	Wisconsin Diagnostic Breast Cancer	WDBC	569 (424)*	30	2
3.	Diabetic Retinopathy Debrecen	MESS	1151 (1080)*	19	2
4.	Cardiotocography	CTG	2126 (528)*	21	3
5.	Ionosphere	IONO	351 (252)*	34	2
6.	Cryotherapy	CRYO	90 (84)*	6	2

* – Wartości w nawiasach wskazują liczbę obiektów w zrównoważonej wersji bazy.

W badaniach eksperymentalnych wykorzystane zostały wyłącznie bazy danych zawiera-

jące dane rzeczywiste. **Tym samym zrealizowany został szósty cel pracy.**
Należy zwrócić uwagę, że bazy użyte do testowania proponowanych w niniejszej rozprawie rozwiązań, w wersji oryginalnej, zawierają silnie niezrównoważone klasy. Bazy 1 – 6 pochodzą z repozytoriów *UC Irvine Machine Learning Repository* i zawierają rzeczywiste dane, bez brakujących wartości w wektorach cech:

a) Baza *Wine* [1] zawiera wyniki analizy chemicznej 3 różnych odmian win pochodzących z jednego rejonu Włoch. Analizie ilościowej podlegało 13 parametrów możliwych do zarejestrowania w każdej z 3 odmian win. Klasa mniejszościowa w tej bazie jest reprezentowana przez 32% mniej obiektów niż klasa najbardziej liczna.
b) Baza *Wisconsin Diagnostic Breast Cancer* [41] zawiera cechy wyznaczone na podstawie analizy zdigitalizowanych obrazów próbek tkanki płucnej pobranych metodą biopsji cienkoigłowej. Charakteryzują one jądra komórkowe widoczne na obrazach. Klasa mniejszościowa w tej bazie jest reprezentowana przez 41% mniej obiektów niż klasa najbardziej liczna.
c) Baza *Diabetic Retinopathy Debrecen* [2] zawiera cechy wyodrębnione z bazy obrazów *Messidor* mające umożliwić rozpoznanie retinopatii cukrzycowej. Cechy reprezentują wykrytą zmianę, charakterystykę cech anatomicznych, oraz deskryptory obrazu. Klasa mniejszościowa w tej bazie jest reprezentowana przez 12% mniej obiektów niż klasa najbardziej liczna.
d) Baza *Cardiotocography* [3] zawiera automatycznie przetworzone wyniki kardiotokogramów płodów (CTG). W wyniku analizy zostały wyodrębnione cechy diagnostyczne. Wynik każdego badania był opisany przez trzech specjalistów jako normalny, podejrzany lub patologiczny. Klasa mniejszościowa w tej bazie jest reprezentowana przez 89% mniej obiektów niż klasa najbardziej liczna.
e) Baza *Ionosphere* [57] zawiera dane pochodzące z radaru meteorologicznego. Cechy opisują część rzeczywistą i urojoną odpowiedzi na każdy z 17 impulsów wysłanych przez radar. Seria impulsów przypisana jest do jednej z dwóch klas, odpowiadającej obecności lub brakowi wolnych elektronów w jonosferze. Klasa mniejszościowa w tej bazie jest reprezentowana przez 54% mniej obiektów niż klasa najbardziej liczna.
f) Baza *Cryotherapy* [34] zawiera dane dotyczące leczenia brodawek, wywołanych zakażeniem wirusem HPV, metodą krioterapii. Obiekty są klasyfikowane według reakcji (lub jej braku) na leczenie. Klasa mniejszościowa liczy 12% mniej obiektów niż klasa bardziej liczna.

W celu uniknięcia błędów implementacyjnych, wykorzystano gotowe implementacje klasyfikatorów k-NN, Wk-NN, DWk-NN oraz RSk-NN dostępne w oprogramowaniu *KNIME* [5], *WEKA* [24] oraz *Matlab*. Prezentowany w Rozprawie Klasyfikator komitetowy SFk-NN/C został zaimplementowany w języku *C#*. Eksperymenty zostały przeprowadzone z użyciem oprogramowania *KNIME*, które pozwala na wykorzystywanie zewnętrznych komponentów.

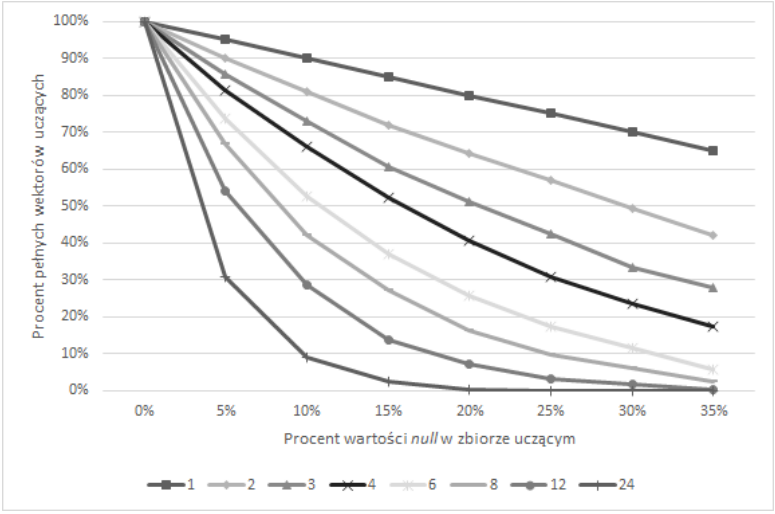
8.1 Uzasadnienie wyboru struktury komitetu klasyfikatorów

W metodach uczenia maszynowego, niektóre rozwiązania wymagają uzupełniania brakujących danych (np. wartością średnią danej cechy) [28, 29, 39, 44, 51, 56], co może zakłócić jakość klasyfikacji. Efekty takiego zakłócenia opisano w Rozdziale 1 (Tabele 1.1 i 1.2), gdzie do uzupełniania wartości brakujących cech zastosowano odpowiednio metodę EM i regresję liniową.

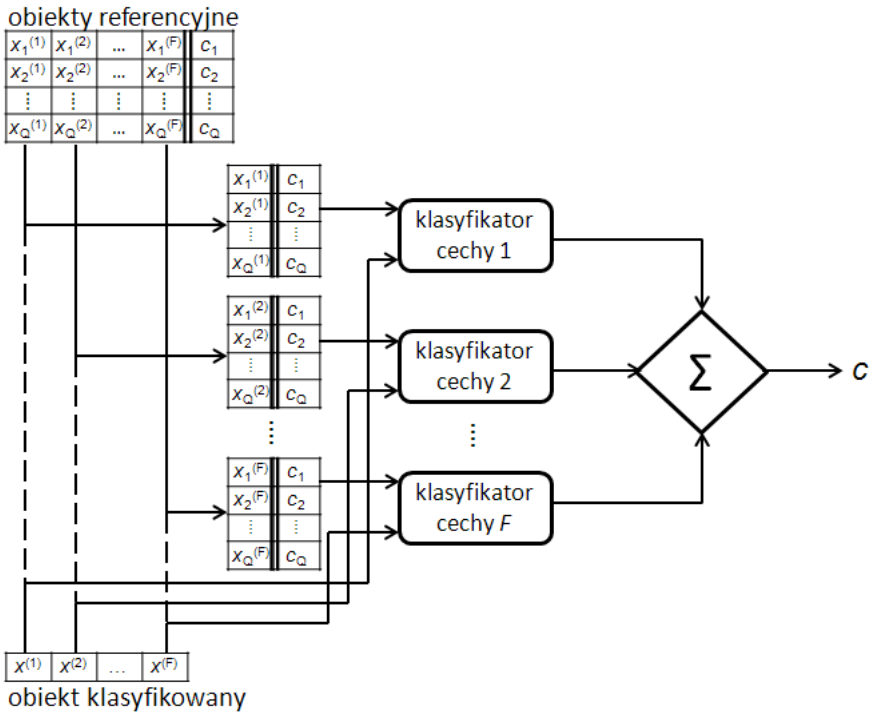
Nadrzędnym zadaniem niniejszej pracy, zawartym w Tezie rozprawy, jest poszukiwanie komitetowych metod klasyfikacji, które nie wymagają uzupełniania brakujących danych (cech) ani na etapie uczenia, ani w trakcie klasyfikacji. Konieczny był zatem wybór struktury komitetu klasyfikatorów adekwatnej do postawionego problemu. Aby uniknąć utraty całych wektorów uczących z powodu wystąpienia w nich wartości *null* należy dokonać podziału przestrzeni cech.

Niech $\mathbf{X}$ będzie w pełni zdefiniowanym zbiorem danych $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{M}$ oraz $\mathbf{x}_i = [x_i^{(1)}, \ldots, x_i^{(F)}]$ jest wektorem F cech, co oznacza, że $\bigwedge_i \bigwedge_j (x_i^{(j)} \neq null)$. Liczba kardynalna zbioru $\mathbf{X}$ wynosi $card\{\mathbf{X}\} = M \cdot F$.
W zbiorze $\mathbf{X}$ zostaje zastąpionych wartością *null* kolejno 5%,10%,...,35% wartości cech. Następnie wszystkie F-elementowe wektory $\mathbf{x}$ dzielone są zgodnie z podziałem π:
π_1: $\mathbf{x} = [x^{(1)}, \ldots, x^{(F)}]$,
π_2: $\mathbf{x}_1 = [x^{(1)}, \ldots, x^{(\lceil F/2 \rceil)}]$, $\mathbf{x}_2 = [x^{(\lceil F/2+1 \rceil)}, \ldots, x^{(F)}]$,
π_3: $\mathbf{x}_1 = [x^{(1)}, \ldots, x^{(\lceil F/3 \rceil)}]$, $\mathbf{x}_2 = [x^{(\lceil F/3+1 \rceil)}, \ldots, x^{(\lceil 2F/3 \rceil)}]$, $\mathbf{x}_3 = [x^{(\lceil 2F/3+1 \rceil)}, \ldots, x^{(F)}]$,
$\vdots$
π_F: $\mathbf{x}_1 = [x^{(1)}]$, $\mathbf{x}_2 = [x^{(2)}], \ldots, \mathbf{x}_M = [x^{(F)}]$.

Zależność liczby pełnych wektorów w funkcji liczby wartości *null* dla różnych podziałów przestrzeni cech można przedstawić w formie wykresu. Wykres przedstawiony na Rys. 8.1 powstał w wyniku podziału przestrzeni cech zbiorów danych opisanych 24 cechami na podzbiory zawierające po 12, 8, 6, 3, 2 i 1 cechę. Pierwotne zbiory danych zawierały różną $(0\%, 5\%, \ldots, 35\%)$ liczbę wartości *null* równomiernie rozłożonych pomiędzy cechami. Wykres przedstawia zależność między długością wektorów na których opisane są podprzestrzenie cech, a liczbą pełnych wektorów. Analiza wykresu potwierdza, że im jest mniej cech w pojedynczym wektorze, tym więcej istnieje wektorów bez wartości *null*. Jeśli pierwotny wektor cech zostanie rozdzielony na wektory jednoelementowe, liczba niemożliwych do wykorzystania wektorów jest najmniejsza i równa liczbie obiektów w których dana cecha przyjmuje wartość *null*. Zatem, aby wykazać słuszność Tezy pracy o możliwości budowy klasyfikatora bez konieczności uzupełniania lub usuwania danych na których operuje klasyfikator, zdecydowano się na strukturę komitetową klasyfikatorów działających w oparciu o pojedyncze cechy obiektów (Rys. 8.2).
Jest to element realizacji drugiego celu pracy.



Rys. 8.1: Liczba możliwych do wykorzystania pełnych wektorów zbioru uczącego w zależności od liczby wartości *null* w tym zbiorze dla różnych przestrzeni $\mathbb{R}^F$, $F = 24, 12, 8, 6, 3, 2, 1$.



Rys. 8.2: Schemat budowy komitetu klasyfikatorów jednocechowych.

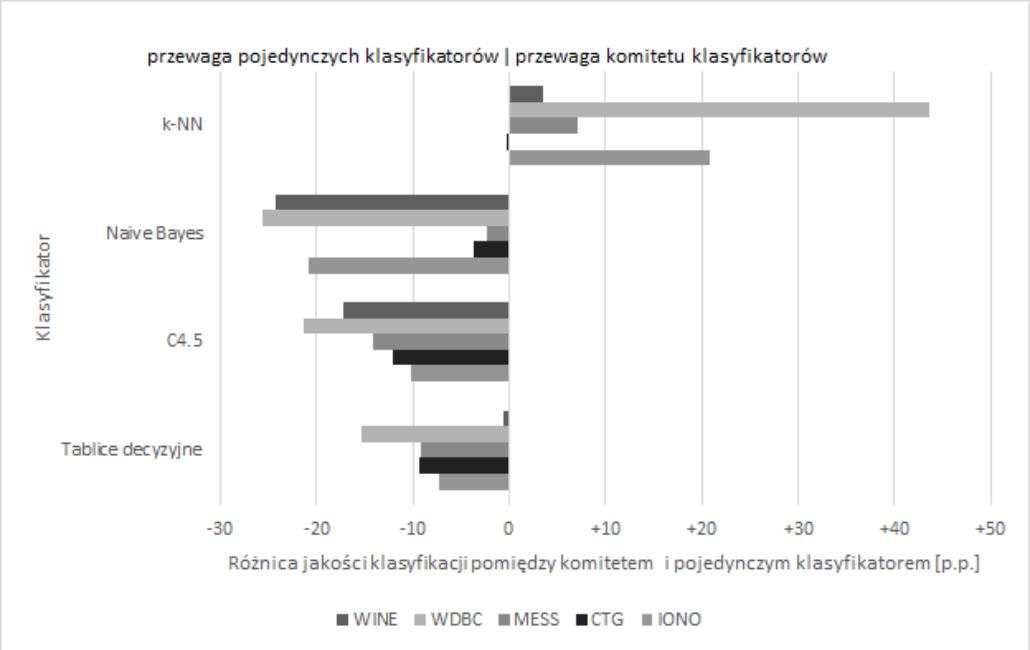
8.2 Uzasadnienie wyboru klasyfikatora użytego do budowy komitetu

Wybór klasyfikatora, który ma zostać użyty w komitecie o strukturze zaproponowanej w Podrozdziale 8.1 poprzedzony był badaniami eksperymentalnymi, z użyciem różnych klasyfikato-

rów i baz danych z Tabeli 8.1. Przeprowadzono testy dla klasyfikatora k-NN (jako przykład klasyfikatorów leniwych), naiwnego klasyfikatora Bayesa (jako przykład klasyfikatorów probabilistycznych), klasyfikatora *C4.5* (jako przykład drzew decyzyjnych) oraz tablic decyzyjnych (jako reprezentanta klasyfikatorów regułowych).

Dla każdej z baz wygenerowano po 10 zbiorów danych, z których w każdym zastąpiono 25% losowo wybranych wartości cech, wartościami *null*. Tak zmodyfikowane dane poddano próbie krzyżowej walidacji z użyciem pojedynczych klasyfikatorów i komitetów zbudowanych z wyżej wymienionych klasyfikatorów działających na pojedynczych cechach obiektów. Na tym etapie badań decyzja klasyfikatora komitetowego wyłaniana była przez proste, większościowe głosowanie decyzji klasyfikatorów wchodzących w skład komitetu.

Wyniki eksperymentu przedstawione na Rysunku 8.3 obrazują różnicę ogólnej jakości klasyfikacji, wyrażoną w punktach procentowych (p.p.), pomiędzy pojedynczymi klasyfikatorami i komitetem zbudowanym z użyciem tego samego klasyfikatora według schematu z Rys. 8.2.



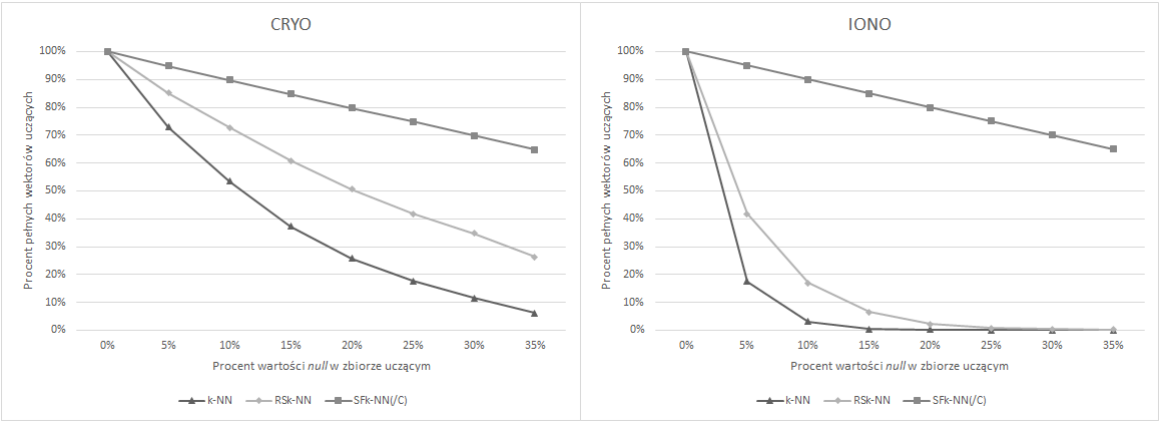
Rys. 8.3: Porównanie jakości klasyfikacji pojedynczych i komitetowych homogenicznych klasyfikatorów jednocechowych w odniesieniu do różnych baz danych.

Jak widać na Rys. 8.3 komitet klasyfikatorów k-NN jako jedyny uzyskał wyższą skuteczność klasyfikacji danych pochodzących ze wszystkich baz w porównaniu z wynikami klasyfikacji otrzymanymi za pomocą pojedynczych klasyfikatorów. W przypadku wszystkich pozostałych testowanych klasyfikatorów, po włączeniu ich w proponowaną strukturę komitetu, następowało pogorszenie ogólnej jakości klasyfikacji. Uwarunkowało to budowę komitetu homogenicznego, zatem tego typu komitety klasyfikatorów k-NN i ich modyfikacje będą podstawą kolejnych eksperymentów.

Jest to realizacja drugiego oraz trzeciego celu pracy,

8.3 Szacowanie liczby traconych wektorów referencyjnych

Występowanie wartości *null* w zbiorach danych przyczynia się do ograniczenia możliwych do wykorzystania wektorów uczących. Zjawisko to ma negatywny wpływ na jakość klasyfikacji. W praktyce stosuje się strategię uzupełniania danych lub realizuje się klasyfikatory, które nie są wrażliwe na obecność danych niekompletnych. W przypadku, gdy nie stosujemy specjalnych struktur klasyfikatorów i nie uzupełniamy danych niepełnych, możemy oszacować liczbę możliwych do wykorzystania, czyli nie zawierających wartości *null*, wektorów referencyjnych. Wektory zawierające wartości *null* muszą zostać usunięte ze zbioru referencyjnego. Jest to niekorzystne, gdyż może prowadzić do drastycznego zmniejszenia liczby elementów zbioru uczącego. Charakter tego zjawiska przedstawiono na Rys. 8.4, gdzie zobrazowano stopień degradacji zbioru referencyjnego $\{\mathbf{x}_q\}_{q=1}^{Q}$ w zależności od liczby niezdefiniowanych cech (=*null*) w wektorach $\mathbf{x}_q$ dla bazy o najmniejszej (*CRYO*, $F = 6$) i największej (*IONO*, $F = 34$) spośród badanych baz, liczbie cech (wykresy przedstawiają uśrednione wyniki dla 10 prób z losowym rozmieszczeniem wartości *null*).



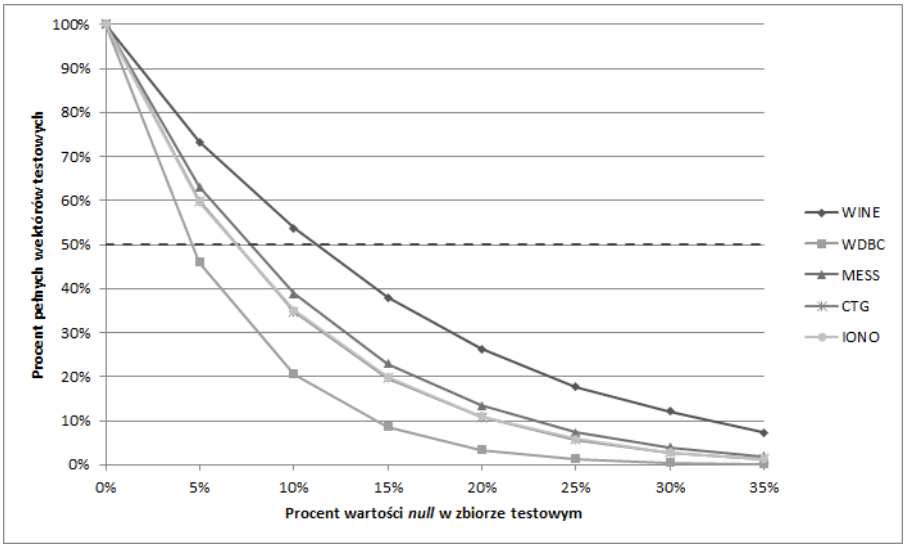
Rys. 8.4: Liczby możliwych do wykorzystania pełnych wektorów ze zbioru uczącego (referencyjnych) w zależności od metody klasyfikacji.

Zjawisko takie występować będzie w klasyfikatorze k-NN oraz jego modyfikacjach (Wk-NN, DWk-NN, EWk-NN, AFFk-NN) i opartych o niego klasyfikatorach komitetowych (RSk-NN, SFk-NN), przedstawionych w Rozdziale 6. Żadna z niekomitetowych modyfikacji metody k-NN nie wpływa na przedstawione zależności, zatem na wykresie ujęte zostały wspólnie, jako k-NN. Obserwacja przebiegu charakterystyk na Rys. 8.4 wskazuje, że opisywana degradacja jest zawsze liniowa dla klasyfikatora SFk-NN (oraz SFk-NN/C) i nieliniowa dla innych klasyfikatorów. Wynika to z faktu, że klasyfikator SFk-NN operuje zawsze na pojedynczych cechach, a pozostałe klasyfikatory posiłkują się zbiorem cech.

Analiza wykresów zaprezentowanych na Rys. 8.4 wskazuje, że proponowany w rozprawie klasyfikator SFk-NN, pozwala na wykorzystywanie największej liczby pełnych wektorów uczących w procesie uczenia klasyfikatorów przy równoczesnej obecności dużej liczby brakujących wartości cech w zbiorze uczącym. Należy uznać to za zaletę proponowanego rozwiązania.
Badania tego typu związane są z realizacją pierwszego celu pracy.

Oszacowanie odporności klasyfikatora RSk-NN na wartości *null* w wektorach klasyfikowanych

Badania klasyfikatora RSk-NN zaprogramowano w sposób następujący. Zgodnie z Algorytmem 5, dla baz z Tabeli 8.1 utworzono $M = 10$ podzbiorów pochodzących z wejściowego zbioru wybranej bazy danych. Badania przeprowadzono odrębnie dla wszystkich baz.
Najpierw ze zbioru cech pierwotnych wektorów $\mathbf{x}_q = [x_q^{(1)}, x_q^{(2)}, \ldots, x_q^{(F)}]$, $q = 1, \ldots, Q$ usuwano kolejno 0%, 5%,...,35% wartości losowo wybranych cech, zastępując je wartościami *null*.
Następnie, dla każdego z M klasyfikatorów, losowo wybrano $r = \lfloor F/2 \rfloor$ cech i utworzono z nich nowe r-elementowe wektory $\tilde{\mathbf{x}}_q^r$.
Na Rys. 8.5 przedstawiono prawdopodobieństwo wylosowania pełnych (nie zawierających wartości *null*) wektorów klasyfikowanych $\mathbf{x}$ dla różnych baz z Tabeli 8.1. W badaniach symulacyjnych wykonano 10 krotne losowanie wektorów $\tilde{\mathbf{x}}_q^r$ i wynik uśredniono.



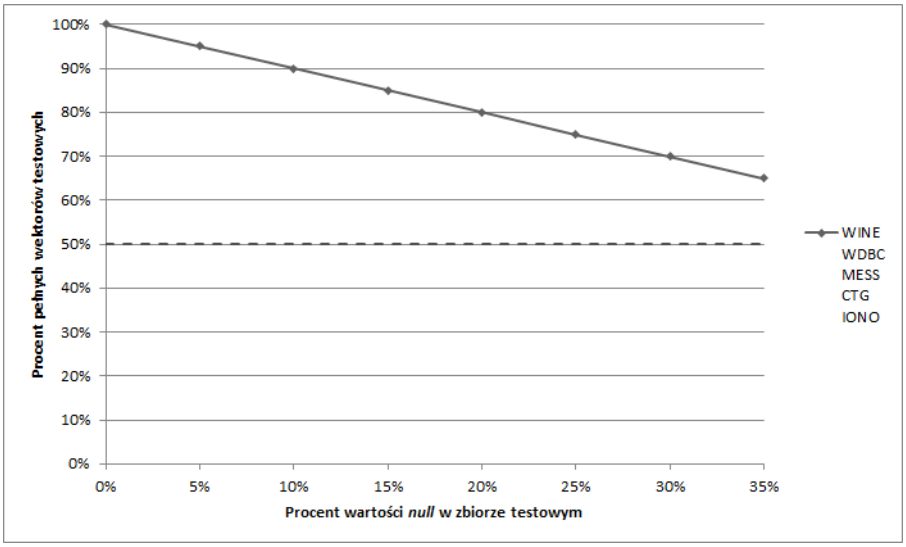
Rys. 8.5: Klasyfikator komitetowy RSk-NN. Liczba możliwych do wykorzystania pełnych wektorów ze zbioru testowego (wektorów klasyfikowanych) w zależności od liczby wartości *null* w tym zbiorze.

Omawiany klasyfikator wykazuje nieliniową zależność liczby pełnych wektorów testowych w zależności od liczby wartości *null* w zbiorze testowym. W zależności od liczby cech opisujących obiekt w bazie zbiorze danych, już w przedziale 5% – 15% wartości *null* w wektorach

testowych, ponad 50% wektorów testowych było niepełnych bez względu na rodzaj bazy, na której przeprowadzono testy.

Oszacowanie odporności klasyfikatora SFk-NN(/C) na wartości *null* w wektorach klasyfikowanych

Badania klasyfikatora SFk-NN(/C) zaprogramowano w sposób analogiczny do badań klasyfikatora RSk-NN. Dla baz z Tabeli 8.1 utworzono F podzbiorów wybranej bazy danych. Badania przeprowadzono odrębnie dla wszystkich baz.
Najpierw ze zbioru cech pierwotnych wektorów $\mathbf{x}_q = [x_q^{(1)}, x_q^{(2)}, \ldots, x_q^{(F)}]$, $q = 1, \ldots, Q$ usuwano kolejno 0%, 5%,...,35% wartości cech, zastępując je wartościami *null*.
Potem, dla każdego z F klasyfikatorów, wybierana jest f-ta cecha i utworzone są z nich nowe 1-elementowe wektory $\tilde{\mathbf{x}}_q^f$.



Rys. 8.6: Klasyfikator komitetowy SFk-NN(/C). Liczba możliwych do wykorzystania pełnych wektorów ze zbioru testowego (wektorów klasyfikowanych) w zależności od liczby wartości *null* w tym zbiorze.

Proponowany w Rozprawie klasyfikator SFk-NN(/C) charakteryzuje się znacznie większą liczbą możliwych do wykorzystania wektorów testowych w porównaniu do omawianego wcześniej klasyfikatora RSk-NN. W analizowanym przedziale wartości *null* w zbiorze testowym, udział pełnych wektorów przekracza 60% i nie zależy od liczby cech obiektów. Jest to zaleta proponowanego rozwiązania. Można zauważyć również liniową zależność udziału pełnych wektorów w stosunku do liczby wartości *null* w zbiorze testowym, co ułatwia antycypowanie zachowań klasyfikatora.

8.4 Wpływ brakujących danych na jakość klasyfikacji

W niniejszej pracy badane są zbiory uczące i testowe, w których występują niepełne dane, co znacząco wpływa na jakość klasyfikacji. Praca klasyfikatora będzie analizowana w zależności od miejsca występowania brakujących danych:

a) dane uczące pozbawione są wartości niektórych cech, co oznacza że cecha w wektorze referencyjnym może posiadać wartość nieokreśloną (*null*),
b) dane testowe pozbawione są wartości niektórych cech, co oznacza że cecha w wektorze klasyfikowanym może posiadać wartość (*null*),
c) zarówno dane uczące jak i testowe pozbawione są wartości niektórych cech.

Przyjęty powyżej podział wynika z rzeczywistych obserwacji, dotyczących kompletności zbiorów danych wejściowych.
W tym Rozdziale przedstawiono wyniki badań eksperymentalnych, których celem było wykazanie wpływu brakujących danych na jakość klasyfikacji, **a zatem realizacja pierwszego celu pracy**. Na potrzeby badań, w oparciu o bazy testowe, w których nie występowały brakujące wartości, stworzone zostały ich zdegradowane wersje z losowo usuniętymi wartościami cech ($x^{(f)}$). Powstały one w wyniku zastąpienia kolejno 5%, 10%, ..., 35% wartości cech, wartością *null*. Badania zostały przeprowadzone dla 3 scenariuszy, odpowiadających wcześniej wymienionym miejscom występowania niepełnych danych:

a) „pełny testowany zdegradowanym" – zbiór uczący oryginalny, zbiór testowy zdegradowany,
b) „zdegradowany testowany pełnym" – zbiór uczący zdegradowany, zbiór testowy oryginalny,
c) „zdegradowany testowany zdegradowanym" – zarówno zbiór uczący jak i testowy zdegradowane.

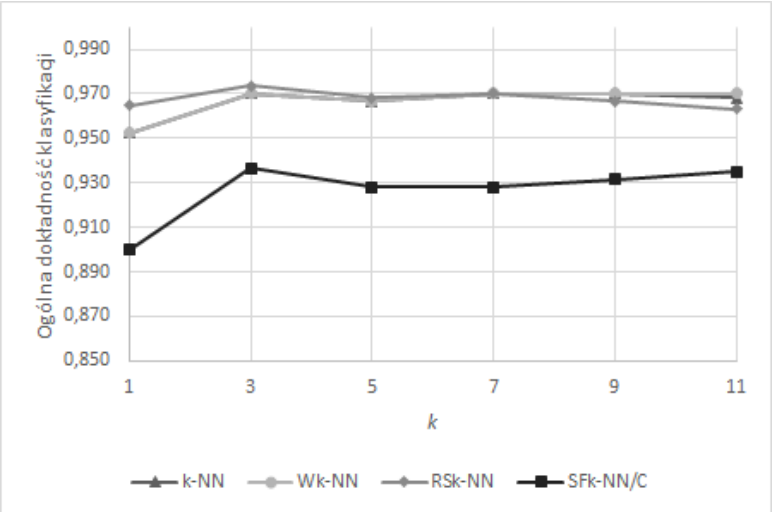
Miarą jakości klasyfikacji była ogólna dokładność klasyfikacji (Tabela 3.2 b)).
Wartości parametrów klasyfikatorów przyjęto zgodnie z Tabelą 8.2.

Tabela 8.2: Wartości parametrów klasyfikatorów.

Klasyfikator	Parametry
k-NN	k = 3
Wk-NN^g	k = 3, w_q zgodnie ze wzorem (6.8)
Wk-NN^l	k = 3, w_q zgodnie ze wzorem (6.9)
DWk-NN	k = 3, w_q zgodnie ze wzorem (6.10)
EWk-NN	k = 3, w_q zgodnie ze wzorem (6.11)
AFFk-NN	k = 3
RSk-NN	k = 3, liczba podprzestrzeni = 10, rozmiar podprzestrzeni = 0,5*
SFk-NN	k = 3
SFk-NN/C	k = 3

* – Domyślne wartości parametrów dla klasyfikatora *Random Subspace* w pakiecie WEKA/KNIME.

Pomimo, że klasyfikator k-NN jest określany jako nieparametryczny [32] analizując jego działanie można zaobserwować wpływ doboru liczby sąsiadów k na jakość klasyfikacji obiektów [30]. Nie ma jednak prostej metody doboru optymalnej wartości parametru k, pozostaje dobór drogą eksperymentalną (Rys. 8.7).



Rys. 8.7: Przykład wpływu parametru k na ogólną dokładność klasyfikacji (baza *WDBC*).

Na potrzeby badań porównawczych różnych klasyfikatorów wartość parametru k nie została dobrana w sposób optymalny, lecz jednakowy dla wszystkich baz i wszystkich klasyfikatorów. Wszystkie porównywane klasyfikatory bazują na metodzie k-NN, zatem na potrzeby porównania tych algorytmów, sprawiedliwym rozwiązaniem wydaje się użycie stałej wartości parametru k.

Poniższe tabele zawierają uśrednione wyniki ogólnej dokładności klasyfikacji (ACC) dla oryginalnych i zdegradowanych baz danych. Dla każdego kroku degradacji przetestowano 10 baz z losowym rozmieszczeniem wartości *null*. Klasyfikacja była wykonywana metodą krzyżowej walidacji typu *leave one out*, według scenariusza „pełny testowany zdegradowanym".

Tabela 8.3: Ogólna dokładność klasyfikacji (ACC) danych z bazy *WINE* według scenariusza „pełny testowany zdegradowanym".

	Procentowa liczba brakujących cech w wektorze danych $\mathbf{x}$								
Klasyfikator	0%	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$
k-NN	0,962	0,931	0,871	0,839	0,798	0,739	0,667	0,631	0,805
Wk-NN^g	0,966	0,933	0,875	0,840	0,801	0,745	0,672	0,634	0,808
Wk-NN^l	0,949	0,844	0,763	0,688	0,616	0,588	0,546	0,500	0,687
DWk-NN	0,949	0,844	0,763	0,688	0,616	0,588	0,546	0,500	0,687
EWk-NN	0,944	0,852	0,772	0,709	0,632	0,610	0,565	0,531	0,702
AFFk-NN	0,962	**0,962**	**0,965**	**0,960**	**0,952**	**0,947**	**0,946**	**0,937**	**0,954**
RSk-NN	**0,972**	0,957	0,912	0,896	0,853	0,798	0,753	0,711	0,857
SFk-NN	0,938	0,940	0,943	0,930	0,928	0,917	0,907	0,888	0,924
SFk-NN/C	0,949	0,943	0,944	0,926	0,932	0,925	0,914	0,897	0,929

Tabela 8.4: Ogólna dokładność klasyfikacji (ACC) danych z bazy *WDBC* według scenariusza „pełny testowany zdegradowanym".

	Procentowa liczba brakujących cech w wektorze danych $\mathbf{x}$								
Klasyfikator	0%	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$
k-NN	0,970	0,894	0,809	0,710	0,630	0,562	0,496	0,453	0,691
Wk-NN^g	0,970	0,894	0,809	0,710	0,630	0,562	0,496	0,453	0,691
Wk-NN^l	0,951	0,926	0,891	0,849	0,804	0,762	0,730	0,699	0,827
DWk-NN	0,951	0,926	0,891	0,849	0,804	0,762	0,730	0,699	0,827
EWk-NN	0,961	0,930	0,891	0,846	0,809	0,763	0,730	0,698	0,829
AFFk-NN	0,970	**0,969**	**0,966**	**0,967**	**0,965**	**0,967**	**0,964**	**0,961**	**0,966**
RSk-NN	**0,974**	0,901	0,822	0,733	0,651	0,573	0,508	0,463	0,703
SFk-NN	0,917	0,917	0,918	0,918	0,918	0,915	0,917	0,914	0,917
SFk-NN/C	0,937	0,936	0,933	0,931	0,928	0,924	0,923	0,918	0,929

Tabela 8.5: Ogólna dokładność klasyfikacji (ACC) danych z bazy *MESS* według scenariusza „pełny testowany zdegradowanym".

	Procentowa liczba brakujących cech w wektorze danych $\mathbf{x}$								
Klasyfikator	0%	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$
k-NN	0,619	0,579	0,562	0,552	0,537	0,536	0,528	0,527	0,555
Wk-NN^g	0,629	0,579	0,562	0,552	0,537	0,536	0,528	0,527	0,556
Wk-NN^l	0,605	0,585	0,571	0,558	0,543	0,539	0,520	0,521	0,555
DWk-NN	0,605	0,585	0,571	0,558	0,543	0,539	0,520	0,521	0,555
EWk-NN	0,599	0,588	0,581	0,557	0,554	0,550	0,539	0,527	0,562
AFFk-NN	0,619	0,619	**0,621**	**0,622**	**0,623**	**0,627**	**0,618**	**0,623**	**0,622**
RSk-NN	**0,674**	**0,627**	0,591	0,568	0,553	0,546	0,540	0,534	0,579
SFk-NN	0,612	0,612	0,612	0,608	0,608	0,607	0,604	0,601	0,608
SFk-NN/C	0,622	0,622	0,617	0,620	0,617	0,610	0,603	0,606	0,615

Tabela 8.6: Ogólna dokładność klasyfikacji (ACC) danych z bazy *CTG* według scenariusza „pełny testowany zdegradowanym".

	Procentowa liczba brakujących cech w wektorze danych $\mathbf{x}$								
Klasyfikator	0%	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$
k-NN	0,913	0,865	0,836	0,811	0,788	0,771	0,763	0,746	0,812
Wk-NN^g	0,918	0,865	0,833	0,805	0,780	0,761	0,751	0,731	0,806
Wk-NN^l	0,919	0,877	0,841	0,802	0,775	0,753	0,734	0,710	0,801
DWk-NN	0,919	0,877	0,841	0,802	0,775	0,753	0,734	0,710	0,801
EWk-NN	0,894	0,867	0,837	0,809	0,783	0,764	0,747	0,729	0,804
AFFk-NN	0,913	**0,911**	**0,909**	**0,908**	**0,908**	**0,903**	**0,904**	**0,901**	**0,907**
RSk-NN	**0,921**	0,885	0,858	0,835	0,817	0,799	0,791	0,783	0,836
SFk-NN	0,778	0,778	0,778	0,778	0,779	0,779	0,779	0,779	0,779
SFk-NN/C	0,870	0,869	0,867	0,869	0,864	0,865	0,864	0,862	0,866

Tabela 8.7: Ogólna dokładność klasyfikacji (ACC) danych z bazy *IONO* według scenariusza „pełny testowany zdegradowanym".

	Procentowa liczba brakujących cech w wektorze danych $\mathbf{x}$								
Klasyfikator	0%	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$
k-NN	0,860	0,851	0,846	0,832	0,805	0,759	0,702	0,635	0,786
Wk-NN^g	0,860	0,851	0,846	0,832	0,805	0,759	0,703	0,635	0,786
Wk-NN^l	0,869	0,855	0,849	0,847	0,838	0,840	0,817	0,801	0,840
DWk-NN	0,869	0,855	0,849	0,847	0,838	0,840	0,817	0,801	0,840
EWk-NN	0,852	0,855	0,849	0,847	0,838	0,840	0,817	0,801	0,837
AFFk-NN	0,860	0,860	0,861	0,861	0,858	0,856	0,858	0,862	0,860
RSk-NN	**0,886**	**0,887**	**0,881**	0,862	0,827	0,775	0,707	0,635	0,808
SFk-NN	0,872	0,875	0,876	**0,877**	0,877	**0,876**	**0,877**	**0,877**	**0,876**
SFk-NN/C	0,875	0,876	0,876	0,876	**0,880**	0,872	0,876	0,873	**0,876**

Tabela 8.8: Ogólna dokładność klasyfikacji (ACC) danych z bazy *CRYO* według scenariusza „pełny testowany zdegradowanym".

	Procentowa liczba brakujących cech w wektorze danych $\mathbf{x}$								
Klasyfikator	0%	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$
k-NN	**0,911**	0,868	0,816	0,798	0,774	0,757	0,710	0,726	0,795
Wk-NN^g	**0,911**	0,868	0,814	0,798	0,774	0,757	0,710	0,726	0,795
Wk-NN^l	0,867	0,878	0,837	0,838	0,793	0,771	0,759	0,740	0,810
DWk-NN	0,867	0,878	0,837	0,838	0,793	0,771	0,759	0,740	0,810
EWk-NN	0,867	0,878	0,837	0,838	0,793	0,771	0,759	0,740	0,810
AFFk-NN	**0,911**	**0,894**	**0,872**	**0,861**	**0,853**	**0,852**	**0,816**	**0,806**	**0,858**
RSk-NN	0,878	0,854	0,834	0,837	0,803	0,789	0,772	0,759	0,816
SFk-NN	0,789	0,787	0,790	0,810	0,788	0,793	0,789	0,770	0,789
SFk-NN/C	0,833	0,823	0,823	0,840	0,814	0,821	0,813	0,799	0,821

Z Tabel 8.3 – 8.8 wynika, że średnia jakość klasyfikacji jest najlepsza dla niekomitetowego klasyfikatora AFFk-NN oraz komitetowego klasyfikatora SFk-NN(/C). Klasyfikator AFFk-NN pomija w wektorach referencyjnych cechy, których wartości nie są znane w wektorze klasyfikowanym. W przypadku klasyfikatora leniwego jest to rozwiązanie bardzo proste do zaimplementowania, gdyż nie występuje w nim etap uczenia i zmiana rozmiaru przestrzeni cech zbioru referencyjnego nie pociąga za sobą konieczności dodatkowych obliczeń, a redukcja cech wręcz upraszcza proces decyzyjny klasyfikatora. Klasyfikator SFk-NN(/C) uzyskuje drugą w kolejności najlepszą dokładność klasyfikacji. Niższa jakość klasyfikacji metodą SFk-NN(/C) wynika z faktu, że ten klasyfikator był projektowany do klasyfikacji danych niekompletnych, wobec

czego nie powinna dziwić nieco niższa jakość klasyfikacji w sytuacji gdy zbiór referencyjny, a zatem większość danych jest kompletna.

W kolejnych tabelach przedstawiono wyniki analogicznego eksperymentu przeprowadzonego według scenariusza „zdegradowany testowany pełnym".

Tabela 8.9: Ogólna dokładność klasyfikacji (ACC) danych z bazy *WINE* według scenariusza „zdegradowany testowany pełnym".

	Procentowa liczba brakujących cech w wektorze danych $\mathbf{x}$								
Klasyfikator	0%	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$
k-NN	0,962	0,957	0,951	0,935	0,913	0,915	0,880	0,849	0,920
Wk-NN^g	0,966	0,957	0,951	0,937	0,915	0,918	0,890	0,862	0,925
Wk-NN^l	0,949	0,951	0,937	0,913	0,884	0,850	0,792	0,748	0,878
DWk-NN	0,949	0,951	0,937	0,913	0,884	0,850	0,792	0,748	0,878
EWk-NN	0,944	0,920	0,908	0,889	0,829	0,794	0,748	0,698	0,841
AFFk-NN	0,962	0,956	0,948	0,921	0,741	0,494	0,329	0,330	0,710
RSk-NN	**0,972**	**0,974**	**0,969**	**0,969**	**0,957**	**0,962**	0,925	0,909	**0,955**
SFk-NN	0,938	0,944	0,948	0,944	0,944	0,945	**0,940**	**0,941**	0,943
SFk-NN/C	0,949	0,944	0,945	0,942	0,941	0,940	0,935	0,937	0,942

Tabela 8.10: Ogólna dokładność klasyfikacji (ACC) danych z bazy *WDBC* według scenariusza „zdegradowany testowany pełnym".

	Procentowa liczba brakujących cech w wektorze danych $\mathbf{x}$								
Klasyfikator	0%	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$
k-NN	0,970	0,956	0,932	0,907	0,896	0,885	0,881	0,820	0,906
Wk-NN^g	0,970	0,956	0,932	0,907	0,896	0,885	0,881	0,820	0,906
Wk-NN^l	0,951	0,950	0,935	0,924	0,903	0,900	0,890	0,877	0,916
DWk-NN	0,951	0,950	0,935	0,924	0,903	0,900	0,890	0,877	0,916
EWk-NN	0,961	0,949	0,935	0,929	0,913	0,907	0,891	0,880	0,921
AFFk-NN	0,970	0,953	0,906	0,704	0,627	0,627	0,627	0,627	0,755
RSk-NN	**0,974**	**0,966**	**0,960**	**0,950**	**0,935**	0,901	0,897	0,855	**0,930**
SFk-NN	0,917	0,918	0,919	0,918	0,916	0,918	0,918	0,919	0,918
SFk-NN/C	0,937	0,932	0,933	0,930	0,928	**0,927**	**0,926**	**0,925**	**0,930**

Tabela 8.11: Ogólna dokładność klasyfikacji (ACC) danych z bazy *MESS* według scenariusza „zdegradowany testowany pełnym".

	Procentowa liczba brakujących cech w wektorze danych $\mathbf{x}$								
Klasyfikator	0%	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$
k-NN	0,619	0,601	0,604	0,562	0,556	0,524	0,518	0,546	0,566
Wk-NN^g	0,629	0,601	0,604	0,562	0,556	0,524	0,518	0,546	0,568
Wk-NN^l	0,605	0,614	0,610	0,599	0,591	0,577	0,551	0,556	0,588
DWk-NN	0,605	0,614	0,610	0,599	0,591	0,577	0,551	0,556	0,588
EWk-NN	0,599	0,616	0,610	0,597	0,592	0,579	0,546	0,555	0,587
AFFk-NN	0,619	0,602	0,597	0,560	0,550	0,501	0,487	0,487	0,550
RSk-NN	**0,674**	**0,660**	**0,656**	**0,636**	**0,628**	0,609	0,590	0,577	**0,629**
SFk-NN	0,612	0,614	0,606	0,619	0,608	0,609	0,609	0,604	0,610
SFk-NN/C	0,622	0,620	0,619	0,620	0,614	**0,621**	**0,613**	**0,611**	0,618

Tabela 8.12: Ogólna dokładność klasyfikacji (ACC) danych z bazy *CTG* według scenariusza „zdegradowany testowany pełnym".

	Procentowa liczba brakujących cech w wektorze danych $\mathbf{x}$								
Klasyfikator	0%	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$
k-NN	0,913	0,888	0,864	0,842	0,818	0,800	0,792	0,796	0,839
Wk-NN^g	0,918	0,891	0,866	0,845	0,823	0,806	0,790	0,793	0,842
Wk-NN^l	0,919	0,905	0,892	0,879	0,864	0,855	0,836	0,825	0,872
DWk-NN	0,919	0,905	0,892	0,879	0,864	0,855	0,836	0,825	0,872
EWk-NN	0,894	0,889	0,877	0,867	0,855	0,848	0,826	0,814	0,859
AFFk-NN	0,913	0,602	0,597	0,560	0,550	0,501	0,487	0,487	0,587
RSk-NN	**0,921**	**0,914**	**0,900**	**0,886**	**0,873**	0,858	0,833	0,807	**0,874**
SFk-NN	0,778	0,778	0,778	0,778	0,778	0,778	0,778	0,778	0,778
SFk-NN/C	0,870	0,868	0,869	0,869	0,869	**0,869**	**0,868**	**0,869**	0,869

Tabela 8.13: Ogólna dokładność klasyfikacji (ACC) danych z bazy *IONO* według scenariusza „zdegradowany testowany pełnym".

	Procentowa liczba brakujących cech w wektorze danych **x**								
Klasyfikator	0%	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$
k-NN	0,860	0,822	0,815	0,778	0,766	0,749	0,760	0,757	0,788
Wk-NN^g	0,860	0,823	0,815	0,778	0,766	0,749	0,760	0,757	0,789
Wk-NN^l	0,869	0,846	0,830	0,825	0,821	0,806	0,801	0,792	0,824
DWk-NN	0,869	0,846	0,830	0,825	0,821	0,806	0,801	0,792	0,824
EWk-NN	0,852	0,844	0,828	0,824	0,820	0,806	0,801	0,792	0,821
AFFk-NN	0,860	0,787	0,647	0,442	0,415	0,359	0,359	0,359	0,529
RSk-NN	**0,886**	0,866	0,849	0,825	0,793	0,776	0,755	0,740	0,811
SFk-NN	0,872	**0,873**	**0,874**	**0,873**	**0,874**	**0,875**	**0,874**	**0,876**	**0,874**
SFk-NN/C	0,875	0,871	0,870	0,869	0,871	0,869	0,866	0,870	0,870

Tabela 8.14: Ogólna dokładność klasyfikacji (ACC) danych z bazy *CRYO* według scenariusza „zdegradowany testowany pełnym".

	Procentowa liczba brakujących cech w wektorze danych **x**								
Klasyfikator	0%	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$
k-NN	**0,911**	**0,878**	0,837	0,780	0,757	0,712	0,732	0,738	0,793
Wk-NN^g	**0,911**	**0,878**	0,837	0,779	0,759	0,712	0,731	0,738	0,793
Wk-NN^l	0,867	0,854	0,841	0,850	0,829	0,794	0,807	0,789	0,829
DWk-NN	0,867	0,854	0,841	0,850	0,829	0,794	0,807	0,789	0,829
EWk-NN	0,867	0,854	0,841	0,850	0,829	0,792	0,807	0,789	0,829
AFFk-NN	**0,911**	0,876	0,837	0,777	0,744	0,668	0,690	0,606	0,763
RSk-NN	0,878	0,870	**0,866**	**0,860**	0,826	0,852	**0,863**	0,823	**0,855**
SFk-NN	0,789	0,807	0,809	0,812	0,812	0,812	0,827	0,811	0,810
SFk-NN/C	0,833	0,852	0,848	0,856	**0,864**	**0,868**	0,856	**0,849**	0,853

Wyniki zamieszczone w Tabelach 8.9 – 8.14 wskazują, że zaproponowany w niniejszej pracy komitetowy klasyfikator SFk-NN(/C) charakteryzuje się lepszą dokładnością klasyfikacji w porównaniu do pozostałych klasyfikatorów minimalno-odległościowych kiedy zbiór uczący zawiera więcej niż 20% (25% dla bazy Wine) brakujących wartości. Przy mniejszej liczbie wartości *null* w zbiorze uczącym lepszą jakość klasyfikacji zapewnia inny klasyfikator komitetowy - RSk-NN. Rozpatrując wszystkie zbiory zdegradowane, klasyfikatory SFk-NN(/C) oraz RSk-NN uzyskują równorzędną dokładność klasyfikacji. Również w tym scenariuszu wyniki dla bazy *Ionosphere* były odmienne (Tab. 8.13), gdyż w przypadku tej bazy najlepszą ogólną dokładność klasyfikacji zapewniał proponowany klasyfikator komitetowy SFk-NN(/C) już od 5% wartości *null*

w wektorach referencyjnych, natomiast klasyfikator RSk-NN, w przedziale 5% – 15% wartości *null* w zbiorze wektorów referencyjnych, uzyskał drugi najlepszy wynik.

W kolejnych tabelach przedstawione zostały wyniki eksperymentu według ostatniego scenariusza – „zdegradowany testowany zdegradowanym".

Tabela 8.15: Ogólna dokładność klasyfikacji (ACC) danych z bazy *WINE* według scenariusza „zdegradowany testowany zdegradowanym".

	Procentowa liczba brakujących cech w wektorze danych **x**								
Klasyfikator	0%	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$
k-NN	0,962	0,925	0,865	0,804	0,750	0,730	0,636	0,635	0,788
Wk-NN^g	0,966	0,926	0,869	0,811	0,761	0,744	0,665	0,655	0,800
Wk-NN^l	0,949	0,887	0,827	0,757	0,687	0,646	0,585	0,562	0,738
DWk-NN	0,949	0,887	0,827	0,757	0,687	0,646	0,585	0,562	0,738
EWk-NN	0,944	0,852	0,769	0,726	0,649	0,622	0,579	0,541	0,710
AFFk-NN	0,962	0,954	**0,944**	0,913	0,836	0,737	0,592	0,533	0,809
RSk-NN	**0,972**	**0,960**	0,897	0,864	0,807	0,774	0,693	0,654	0,828
SFk-NN	0,938	0,942	0,934	**0,927**	**0,929**	0,912	0,897	0,888	0,921
SFk-NN/C	0,949	0,940	0,943	0,926	0,922	**0,916**	**0,907**	**0,890**	**0,924**

Tabela 8.16: Ogólna dokładność klasyfikacji (ACC) danych z bazy *WDBC* według scenariusza „zdegradowany testowany zdegradowanym".

	Procentowa liczba brakujących cech w wektorze danych **x**								
Klasyfikator	0%	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$
k-NN	0,970	0,892	0,811	0,749	0,677	0,636	0,581	0,621	0,742
Wk-NN^g	0,970	0,892	0,811	0,749	0,677	0,636	0,581	0,621	0,742
Wk-NN^l	0,951	0,923	0,893	0,862	0,843	0,808	0,788	0,764	0,854
DWk-NN	0,951	0,923	0,893	0,862	0,843	0,808	0,788	0,764	0,854
EWk-NN	0,961	0,919	0,885	0,862	0,828	0,803	0,780	0,753	0,849
AFFk-NN	0,970	**0,953**	0,919	0,822	0,625	0,611	0,607	0,616	0,765
RSk-NN	**0,974**	0,898	0,829	0,757	0,705	0,644	0,564	0,599	0,746
SFk-NN	0,917	0,919	0,919	0,916	0,915	0,915	0,916	**0,912**	0,916
SFk-NN/C	0,937	0,932	**0,931**	**0,925**	**0,923**	**0,923**	**0,921**	0,910	**0,925**

Tabela 8.17: Ogólna dokładność klasyfikacji (ACC) danych z bazy *MESS* według scenariusza „zdegradowany testowany zdegradowanym".

	Procentowa liczba brakujących cech w wektorze danych $\mathbf{x}$								
Klasyfikator	0%	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$
k-NN	0,619	0,576	0,568	0,550	0,537	0,527	0,527	0,526	0,554
Wk-NN^g	0,629	0,576	0,567	0,550	0,537	0,527	0,527	0,526	0,555
Wk-NN^l	0,605	0,598	0,582	0,557	0,541	0,539	0,540	0,516	0,560
DWk-NN	0,605	0,598	0,582	0,557	0,541	0,539	0,540	0,516	0,560
EWk-NN	0,599	0,593	0,577	0,552	0,538	0,533	0,536	0,518	0,556
AFFk-NN	0,619	0,603	0,600	0,567	0,562	0,524	0,519	0,516	0,564
RSk-NN	**0,674**	**0,621**	0,584	0,565	0,556	0,548	0,543	0,537	0,579
SFk-NN	0,612	0,614	0,604	0,607	0,605	0,600	0,599	0,590	0,604
SFk-NN/C	0,622	**0,621**	**0,611**	**0,616**	**0,611**	**0,614**	**0,602**	**0,597**	**0,612**

Tabela 8.18: Ogólna dokładność klasyfikacji (ACC) danych z bazy *CTG* według scenariusza „zdegradowany testowany zdegradowanym".

	Procentowa liczba brakujących cech w wektorze danych $\mathbf{x}$								
Klasyfikator	0%	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$
k-NN	0,913	0,857	0,820	0,797	0,790	0,780	0,772	0,773	0,813
Wk-NN^g	0,918	0,860	0,819	0,795	0,788	0,776	0,763	0,768	0,811
Wk-NN^l	0,919	0,875	0,849	0,821	0,803	0,784	0,765	0,748	0,821
DWk-NN	0,919	0,875	0,849	0,821	0,803	0,784	0,765	0,748	0,821
EWk-NN	0,894	0,865	0,840	0,819	0,801	0,785	0,770	0,761	0,817
AFFk-NN	0,913	**0,887**	0,863	0,840	0,811	0,791	0,760	0,757	0,828
RSk-NN	**0,921**	0,880	0,843	0,819	0,801	0,793	0,785	0,780	0,828
SFk-NN	0,778	0,778	0,778	0,778	0,779	0,779	0,779	0,779	0,779
SFk-NN/C	0,870	0,868	**0,868**	**0,870**	**0,865**	**0,864**	**0,862**	**0,859**	**0,866**

Tabela 8.19: Ogólna dokładność klasyfikacji (ACC) danych z bazy *IONO* według scenariusza „zdegradowany testowany zdegradowanym".

	Procentowa liczba brakujących cech w wektorze danych **x**								
Klasyfikator	0%	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$
k-NN	0,860	0,818	0,815	0,745	0,716	0,712	0,652	0,607	0,741
Wk-NN^g	0,860	0,818	0,815	0,744	0,716	0,712	0,652	0,607	0,741
Wk-NN^l	0,869	0,852	0,842	0,837	0,821	0,806	0,798	0,785	0,826
DWk-NN	0,869	0,852	0,842	0,837	0,821	0,806	0,798	0,785	0,826
EWk-NN	0,852	0,838	0,819	0,818	0,814	0,800	0,794	0,787	0,815
AFFk-NN	0,860	0,796	0,686	0,539	0,476	0,389	0,375	0,365	0,561
RSk-NN	**0,886**	0,870	0,847	0,810	0,763	0,743	0,713	0,687	0,790
SFk-NN	0,872	**0,875**	**0,876**	**0,874**	0,874	**0,877**	**0,874**	**0,877**	**0,875**
SFk-NN/C	0,875	0,874	0,873	0,872	**0,877**	0,870	0,872	0,873	0,873

Tabela 8.20: Ogólna dokładność klasyfikacji (ACC) danych z bazy *CRYO* według scenariusza „zdegradowany testowany zdegradowanym".

	Procentowa liczba brakujących cech w wektorze danych **x**								
Klasyfikator	0%	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$
k-NN	**0,911**	0,843	0,788	0,726	0,743	0,690	0,692	0,698	0,761
Wk-NN^g	**0,911**	0,843	0,788	0,726	0,743	0,690	0,693	0,697	0,761
Wk-NN^l	0,867	0,824	0,788	0,786	0,757	0,723	0,703	0,703	0,769
DWk-NN	0,867	0,824	0,788	0,786	0,757	0,723	0,703	0,703	0,769
EWk-NN	0,867	0,776	0,767	0,759	0,748	0,710	0,696	0,694	0,752
AFFk-NN	**0,911**	**0,866**	0,828	0,779	0,766	0,750	0,731	0,697	0,791
RSk-NN	0,878	0,841	0,827	0,802	0,744	0,761	0,730	0,736	0,790
SFk-NN	0,789	0,797	0,802	0,814	0,784	0,794	0,799	0,776	0,794
SFk-NN/C	0,833	0,838	**0,830**	**0,841**	**0,824**	**0,828**	**0,814**	**0,781**	**0,824**

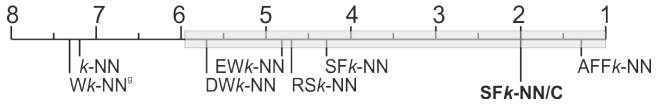
Analiza wyników zaprezentowanych w Tabelach 8.15 – 8.20 potwierdza, że proponowane klasyfikatory komitetowe SFk-NN oraz SFk-NN/C uzyskują średnio lepszą dokładność klasyfikacji niż pozostałe minimalno-odległościowe klasyfikatory leniwe kiedy zbiory uczący i testowy zawierają 10% (5% dla bazy *Ionosphere*) lub więcej brakujących wartości.

W przypadku uczenia klasyfikatora na pełnych danych i wartościach *null* występujących jedynie w danych klasyfikowanych, lepszą jakość klasyfikacji można uzyskać stosując zwykły klasyfikatora k-NN w połączeniu z filtrowaniem cech, których wartości były nieoznaczone. Liczba brakujących danych jest znana *a priori*, przed rozpoczęciem klasyfikacji, zatem w oparciu o tę informację można podjąć decyzję o wyborze odpowiedniego klasyfikatora.

Aby lepiej uwidocznić różnice w jakości klasyfikacji osiąganej przez klasyfikatory, w oparciu o dane z Tabel 8.3 – 8.8, 8.9 – 8.14 oraz 8.15 – 8.20, utworzono rankingi dla każdego ze scenariuszy testowych. Rankingi te obrazują średnią pozycję klasyfikatorów pod względem dokładności klasyfikacji z wyróżnieniem poszczególnych stopni degradacji danych. Rankingi zostały przedstawione w Tabelach 8.21 – 8.23. Poniżej każdej tabeli zamieszczony został Rysunek będący graficzną reprezentacją rankingu uśrednionych wyników ogólnej jakości klasyfikacji danych niepełnych uzyskiwanych przez poszczególne klasyfikatory. Dodatkowo zaznaczony został przedział odzwierciedlający różnicę krytyczną, czyli minimalną różnicę rangi, świadczącą o statystycznej istotnej różnicy pomiędzy porównywanymi klasyfikatorami. Różnica krytyczna wyznaczona była w oparciu o *test Bonferroniego-Dunna* dla α = 0,10.

Tabela 8.21: Ranking dokładności klasyfikacji według scenariusza „pełny testowany zdegradowanym" na danych zawierających 5% – 35% wartości *null*.

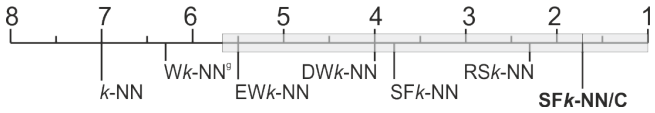
Baza	*k*-NN	W*k*-NN^g	W*k*-NN^l	DW*k*-NN	EW*k*-NN	AFF*k*-NN	RS*k*-NN	SF*k*-NN	SF*k*-NN/C
WINE	0,782	0,786	0,649	0,649	0,667	0,953	0,840	0,922	0,926
WDBC	0,651	0,651	0,809	0,809	0,810	0,966	0,664	0,917	0,928
MESS	0,546	0,546	0,548	0,548	0,557	0,622	0,566	0,607	0,614
CTG	0,797	0,789	0,785	0,785	0,791	0,906	0,824	0,779	0,866
IONO	0,776	0,776	0,835	0,835	0,835	0,859	0,796	0,876	0,876
CRYO	0,778	0,778	0,802	0,802	0,802	0,851	0,807	0,790	0,819
Ranga	7,2	7,3	5,7	5,7	4,8	1,3	4,7	4,3	2,0
Pozycja	8	9	6	6	5	1	4	3	2



Rys. 8.8: Graficzna reprezentacja rankingu dla scenariusza „pełny testowany zdegradowanym" z oznaczoną różnicą krytyczną według testu Bonferroniego-Dunna dla α = 0,10.

Tabela 8.22: Ranking dokładności klasyfikacji według scenariusza „zdegradowany testowany pełnym" na danych zawierających 5% – 35% wartości *null*.

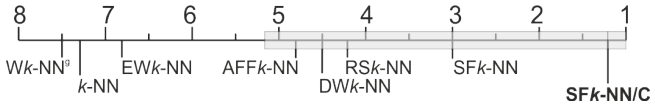
Baza	*k*-NN	W*k*-NN^g	W*k*-NN^l	DW*k*-NN	EW*k*-NN	AFF*k*-NN	RS*k*-NN	SF*k*-NN	SF*k*-NN/C
WINE	0,914	0,919	0,868	0,868	0,827	0,674	0,952	0,944	0,941
WDBC	0,897	0,897	0,911	0,911	0,915	0,724	0,923	0,918	0,929
MESS	0,559	0,559	0,585	0,585	0,585	0,541	0,622	0,610	0,617
CTG	0,829	0,831	0,865	0,865	0,854	0,541	0,867	0,778	0,869
IONO	0,778	0,778	0,817	0,817	0,816	0,481	0,801	0,874	0,869
CRYO	0,776	0,776	0,823	0,823	0,823	0,742	0,851	0,813	0,856
Ranga	7,0	6,3	4,0	4,0	5,5	9,0	2,3	3,8	1,7
Pozycja	8	7	4	4	6	9	2	3	1



Rys. 8.9: Graficzna reprezentacja rankingu dla scenariusza „zdegradowany testowany pełnym" z oznaczoną różnicą krytyczną według testu Bonferroniego-Dunna dla α = 0,10.

Tabela 8.23: Ranking dokładności klasyfikacji według scenariusza „zdegradowany testowany zdegradowanym" na danych zawierających 5% – 35% wartości *null*.

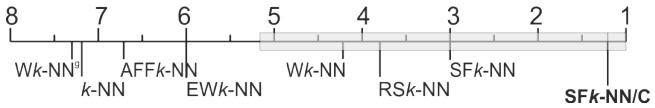
Baza	k-NN	Wk-NN^g	Wk-NN^l	DWk-NN	EWk-NN	AFFk-NN	RSk-NN	SFk-NN	SFk-NN/C
WINE	0,764	0,776	0,707	0,707	0,677	0,787	0,807	0,918	0,921
WDBC	0,710	0,710	0,840	0,840	0,833	0,736	0,714	0,916	0,924
MESS	0,544	0,544	0,553	0,553	0,550	0,556	0,565	0,603	0,610
CTG	0,798	0,796	0,806	0,806	0,806	0,816	0,814	0,779	0,865
IONO	0,724	0,723	0,820	0,820	0,810	0,518	0,776	0,875	0,873
CRYO	0,740	0,740	0,755	0,755	0,736	0,774	0,777	0,795	0,822
Ranga	7,3	7,5	4,5	4,5	6,8	4,8	4,2	3,0	1,2
Pozycja	8	9	4	4	7	6	3	2	1



Rys. 8.10: Graficzna reprezentacja rankingu dla scenariusza „zdegradowany testowany zdegradowanym" z oznaczoną różnicą krytyczną według testu Bonferroniego-Dunna dla α = 0,10.

Tabela 8.24: Ranking uśrednionej dokładności klasyfikacji dla wszystkich scenariuszy na danych zawierających 5% – 35% wartości *null*.

Baza	k-NN	Wk-NN^g	Wk-NN^l	DWk-NN	EWk-NN	AFFk-NN	RSk-NN	SFk-NN	SFk-NN/C
WINE	0,820	0,827	0,741	0,741	0,724	0,805	0,866	0,928	0,929
WDBC	0,752	0,752	0,853	0,853	0,852	0,809	0,767	0,917	0,927
MESS	0,550	0,550	0,562	0,562	0,564	0,573	0,584	0,607	0,614
CTG	0,808	0,805	0,819	0,819	0,817	0,754	0,835	0,778	0,867
IONO	0,759	0,759	0,824	0,824	0,821	0,620	0,791	0,875	0,873
CRYO	0,765	0,765	0,794	0,794	0,787	0,789	0,812	0,799	0,833
Ranga	7,2	7,3	4,3	4,3	6,0	6,7	3,8	3,0	1,2
Pozycja	8	9	4	4	6	7	3	2	1



Rys. 8.11: Graficzna reprezentacja rankingu uśrednionej dokładności klasyfikacji z oznaczoną różnicą krytyczną według testu Bonferroniego-Dunna dla α = 0,10.

Pierwsza pozycja klasyfikatora w zbiorczym rankingu przedstawionym w Tabeli 8.24 dowodzi uniwersalności klasyfikatora SFk-NN/C. Na podstawie testu Bonferroniego-Dunna [13]

można też stwierdzić, że proponowany klasyfikator SFk-NN/C, jako jedyny, niezależnie od scenariusza testowego uzyskuje istotnie lepszą dokładność klasyfikacji niż klasyfikatory k-NN oraz Wk-NN^g. W żadnym ze scenariuszy testowych nie okazał się też istotnie gorszy od żadnego z klasyfikatorów, z którymi został porównany.

Należy zauważyć, że proponowane w rozprawie klasyfikatory SFk-NN/(C) charakteryzują się stałym poziomem ogólnej jakości klasyfikacji w szerokim zakresie degradacji danych. Proponowany klasyfikator SFk-NN(/C) pracując w obecności brakujących wartości cech (do 35%) utrzymywał zdolność prawidłowej klasyfikacji danych, czym wyróżniał się na tle innych klasyfikatorów leniwych – charakteryzował się najlepszą stałością ogólnej jakości klasyfikacji w funkcji liczby brakujących cech, co znajduje potwierdzenie w Tabelach 8.25 – 8.27. Tabele te przedstawiają wariancję:

$$s^2 = \frac{1}{n}\sum_{1}^{n}\left(ACC_i - \overline{ACC}\right)^2, \tag{8.1}$$

gdzie $n = 7$ (obliczenia dla 5%, 10%, 15%, 20%, 25%, 30%, 35% wartości brakujących cech w zbiorze danych), oraz rozstęp *ogólnej jakości klasyfikacji* dla baz danych z 5% – 35% brakujących wartości cech:

$$R = (ACC_{max} - ACC_{min}). \tag{8.2}$$

Tabela 8.25: Wariancja i zakres zmienności ogólnej dokładności klasyfikacji (ACC) dla scenariusza „oryginalny testowany zdegradowanym".

	WINE		WDBC		MESS		CTG		IONO		CRYO	
Klasyfikator	s^2	R	s^2	R	s^2	R	s^2	R	s^2	R	s^2	R
k-NN	0,012	0,300	0,027	0,441	**0,000**	0,052	0,002	0,119	0,007	0,216	0,003	0,158
Wk-NN^g	0,012	0,299	0,027	0,441	**0,000**	0,052	0,002	0,134	0,007	0,216	0,003	0,158
Wk-NN^l	0,015	0,344	0,007	0,227	0,001	0,065	0,004	0,167	**0,000**	0,054	0,002	0,138
DWk-NN	0,015	0,344	0,007	0,227	0,001	0,065	0,004	0,167	**0,000**	0,054	0,002	0,138
EWk-NN	0,013	0,321	0,007	0,232	**0,000**	0,061	0,002	0,138	**0,000**	0,054	0,002	0,138
AFFk-NN	**0,000**	0,028	**0,000**	0,008	**0,000**	0,009	**0,000**	0,010	**0,000**	0,006	0,001	0,089
RSk-NN	0,008	0,246	0,027	0,438	0,001	0,093	0,001	0,102	0,009	0,252	0,001	0,096
SFk-NN	**0,000**	0,055	**0,000**	**0,004**	**0,000**	**0,011**	**0,000**	**0,001**	**0,000**	**0,002**	**0,000**	**0,040**
SFk-NN/C	**0,000**	**0,047**	**0,000**	0,018	**0,000**	0,019	**0,000**	0,007	**0,000**	0,008	**0,000**	0,041

Tabela 8.26: Wariancja i zakres zmienności ogólnej dokładności klasyfikacji (ACC) dla scenariusza „zdegradowany testowany oryginalnym".

	WINE		WDBC		MESS		CTG		IONO		CRYO	
Klasyfikator	s^2	R	s^2	R	s^2	R	s^2	R	s^2	R	s^2	R
k-NN	0,002	0,108	0,002	0,136	0,001	0,086	0,001	0,096	0,001	0,073	0,004	0,166
Wk-NN^g	0,001	0,095	0,002	0,136	0,001	0,086	0,001	0,101	0,001	0,074	0,004	0,166
Wk-NN^l	0,006	0,203	0,001	0,073	0,001	0,063	0,001	0,080	**0,000**	0,054	0,001	0,066
DWk-NN	0,006	0,203	0,001	0,073	0,001	0,063	0,001	0,080	**0,000**	0,054	0,001	0,066
EWk-NN	0,007	0,222	0,001	0,069	0,001	0,070	0,001	0,075	**0,000**	0,052	0,001	0,066
AFFk-NN	0,082	0,627	0,021	0,326	0,002	0,115	0,002	0,115	0,029	0,428	0,009	0,270
RSk-NN	0,001	0,065	0,002	0,111	0,001	0,083	0,001	0,107	0,002	0,126	**0,000**	0,047
SFk-NN	**0,000**	**0,008**	**0,000**	**0,003**	**0,000**	0,015	**0,000**	**0,000**	**0,000**	**0,003**	**0,000**	**0,020**
SFk-NN/C	**0,000**	0,010	**0,000**	0,008	**0,000**	**0,010**	**0,000**	0,001	**0,000**	0,005	**0,000**	**0,020**

Tabela 8.27: Wariancja i zakres zmienności ogólnej dokładności klasyfikacji (ACC) dla scenariusza „zdegradowany testowany zdegradowanym".

	WINE		WDBC		MESS		CTG		IONO		CRYO	
Klasyfikator	s^2	R	s^2	R	s^2	R	s^2	R	s^2	R	s^2	R
k-NN	0,012	0,290	0,013	0,311	**0,000**	0,050	0,001	0,085	0,006	0,211	0,003	0,153
Wk-NN^g	0,010	0,271	0,013	0,311	**0,000**	0,050	0,001	0,097	0,006	0,211	0,003	0,153
Wk-NN^l	0,015	0,325	0,003	0,159	0,001	0,082	0,002	0,127	0,001	0,067	0,002	0,121
DWk-NN	0,015	0,325	0,003	0,159	0,001	0,082	0,002	0,127	0,001	0,067	0,002	0,121
EWk-NN	0,012	0,311	0,004	0,166	0,001	0,075	0,001	0,104	0,000	0,051	0,001	0,081
AFFk-NN	0,029	0,421	0,024	0,346	0,001	0,087	0,003	0,130	0,028	0,431	0,003	0,169
RSk-NN	0,012	0,306	0,015	0,334	0,001	0,084	0,001	0,100	0,005	0,183	0,002	0,111
SFk-NN	**0,000**	0,054	**0,000**	**0,007**	**0,000**	**0,024**	**0,000**	**0,001**	**0,000**	**0,003**	**0,000**	**0,039**
SFk-NN/C	**0,000**	**0,053**	**0,000**	0,022	**0,000**	**0,024**	**0,000**	0,011	**0,000**	0,007	**0,000**	0,060

Jak wynika z powyższych tabel, wariancja ogólnej jakości klasyfikacji w przypadku klasyfikatorów SFk-NN oraz SFk-NN/C w żadnym przypadku nie przekroczyła 0,4‰. Nawet w przypadku baz, dla których jakość klasyfikacji proponowanego klasyfikatora SFk-NN(/C) nie była najlepsza, była najbardziej stabilna w badanym zakresie degradacji danych.

8.5 Wpływ niezrównoważenia klas na jakość klasyfikacji

W poprzednich eksperymentach, niezrównoważenie klas w bazach nie było zmieniane i wynikało z naturalnej budowy baz danych. W prezentowanym w tym Rozdziale eksperymencie wykorzystano zrównoważone metodą losowego próbkowania bazy wymienione w Tabeli 8.1. Każda z tych baz została sztucznie zakłócona przez losowe usunięcie 25%, 50% lub 75% obiektów (wektorów $\mathbf{x}$) jednej z klas. Eksperyment był powtarzany dla każdej z klas wyróżnionych w bazie. Bazy posiadały losowo usunięte wartości cech (*null*), wygenerowane analogicznie jak w poprzednich eksperymentach (podrozdział 8.4). W ten sposób powstawały zdegradowane bazy z jedną klasą mniejszościową.

Zbiory uczący i testowy były wyłaniane metodą Q-krotnej krzyżowej walidacji (*„leave one out"*), gdzie Q jest liczbą obiektów w bazie danych. Miarą jakości klasyfikacji była *ogólna dokładność* (*ACC*). Eksperymenty zostały przeprowadzone dla wszystkich opisanych w pracy klasyfikatorów, a wyniki badań zebrane zostały w zestawie Tabel 8.28 – 8.32.

Tabela 8.28: Ogólna dokładność klasyfikacji (ACC) danych z bazy *WINE* dla różnych stopni niezbilansowania klas, z brakującymi wartościami.

	Stopień redukcji klasy mniejszościowej			
Klasyfikator	0%	25%	50%	75%
k-NN	0,792	0,807	0,802	0,821
Wk-NN^g	0,800	0,816	0,813	0,832
Wk-NN^l	0,704	0,690	0,701	0,728
DWk-NN	0,704	0,690	0,701	0,728
EWk-NN	0,685	0,686	0,688	0,713
AFFk-NN	0,735	0,714	0,720	0,732
RSk-NN	0,815	0,819	0,832	0,842
SFk-NN	0,918	0,907	0,886	0,873
SFk-NN/C	**0,922**	**0,920**	**0,922**	**0,915**

Tabela 8.29: Ogólna dokładność klasyfikacji (ACC) danych z bazy *WDBC* dla różnych stopni niezbilansowania klas, z brakującymi wartościami.

	Stopień redukcji klasy mniejszościowej			
Klasyfikator	0%	25%	50%	75%
k-NN	0,960	**0,961**	**0,962**	0,962
Wk-NN^g	0,960	**0,961**	**0,962**	0,962
Wk-NN^l	0,948	0,948	0,950	0,953
DWk-NN	0,948	0,948	0,950	0,953
EWk-NN	0,956	0,952	0,953	0,953
AFFk-NN	0,961	**0,961**	**0,962**	0,962
RSk-NN	**0,964**	**0,961**	**0,962**	**0,963**
SFk-NN	0,937	0,922	0,901	0,873
SFk-NN/C	0,939	0,930	0,910	0,873

Tabela 8.30: Ogólna dokładność klasyfikacji (ACC) danych z bazy *MESS* dla różnych stopni niezbilansowania klas, z brakującymi wartościami.

	Stopień redukcji klasy mniejszościowej			
Klasyfikator	0%	25%	50%	75%
k-NN	0,543	0,552	0,605	0,741
Wk-NN^g	0,543	0,552	0,605	0,741
Wk-NN^l	0,556	0,570	0,587	0,702
DWk-NN	0,556	0,569	0,587	0,702
EWk-NN	0,550	0,553	0,603	0,730
AFFk-NN	0,561	0,565	0,616	0,730
RSk-NN	0,552	0,563	0,639	0,779
SFk-NN	0,610	**0,617**	**0,678**	**0,801**
SFk-NN/C	**0,612**	0,601	0,594	0,582

Tabela 8.31: Ogólna dokładność klasyfikacji (ACC) danych z bazy *CTG* dla różnych stopni niezbilansowania klas, z brakującymi wartościami.

	Stopień redukcji klasy mniejszościowej			
Klasyfikator	0%	25%	50%	75%
k-NN	0,605	0,606	0,622	0,654
Wk-NN^g	0,611	0,611	0,625	0,657
Wk-NN^l	0,637	0,642	0,642	0,679
DWk-NN	0,637	0,642	0,642	0,679
EWk-NN	0,620	0,625	0,636	0,668
AFFk-NN	0,564	0,565	0,564	0,603
RSk-NN	0,647	0,651	0,655	0,691
SFk-NN	0,783	0,765	0,740	0,777
SFk-NN/C	**0,785**	**0,782**	**0,776**	**0,793**

Tabela 8.32: Ogólna dokładność klasyfikacji (ACC) danych z bazy *IONO* dla różnych stopni niezbilansowania klas, z brakującymi wartościami.

	Stopień redukcji klasy mniejszościowej			
Klasyfikator	0%	25%	50%	75%
k-NN	0,726	0,693	0,707	0,796
Wk-NN^g	0,726	0,693	0,707	0,796
Wk-NN^l	0,777	0,780	0,779	0,814
DWk-NN	0,777	0,780	0,779	0,814
EWk-NN	0,775	0,782	0,772	0,800
AFFk-NN	0,525	0,529	0,532	0,594
RSk-NN	0,740	0,717	0,755	0,813
SFk-NN	0,855	**0,866**	**0,865**	**0,870**
SFk-NN/C	**0,856**	0,861	0,849	0,810

Tabela 8.33: Ogólna dokładność klasyfikacji (ACC) danych z bazy *CRYO* dla różnych stopni niezbilansowania klas, z brakującymi wartościami.

	Stopień redukcji klasy mniejszościowej			
Klasyfikator	0%	25%	50%	75%
k-NN	0,733	0,738	0,752	0,813
Wk-NN^g	0,733	0,738	0,752	0,813
Wk-NN^l	0,773	0,759	0,774	0,835
DWk-NN	0,773	0,759	0,774	0,835
EWk-NN	0,777	0,756	0,782	0,815
AFFk-NN	0,793	0,769	0,777	0,827
RSk-NN	0,787	0,763	0,794	**0,839**
SFk-NN	0,808	0,792	0,779	0,817
SFk-NN/C	**0,826**	**0,813**	**0,797**	0,727

Zarówno na danych zrównoważonych jak i niezrównoważonych w większości przypadków proponowany klasyfikator SFk-NN lub jego wariant SFk-NN/C uzyskiwały najlepszą ogólną dokładność klasyfikacji. Można to zobrazować posługując się rankingiem jakości klasyfikacji (Tabela 8.34).

Tabela 8.34: Ranking dokładności klasyfikacji dla różnych stopni niezbilansowania klas, z brakującymi wartościami.

	Stopień redukcji klasy mniejszościowej				Średnia	
Klasyfikator	0%	25%	50%	75%	0%–75%	25%–75%
k-NN	6,5	6,2	5,7	5,5	6,0	5,8
Wk-NN^g	6,2	5,8	5,3	5,2	5,6	5,4
Wk-NN^l	5,0	4,8	5,7	4,5	5,0	5,0
DWk-NN	5,0	5,0	5,7	4,5	5,0	5,1
EWk-NN	6,2	6,2	5,7	6,3	6,1	6,1
AFFk-NN	5,5	6,2	5,0	5,3	5,5	5,5
RSk-NN	3,8	3,8	**3,0**	**2,3**	3,3	**3,1**
SFk-NN	3,2	2,8	3,2	3,2	**3,1**	**3,1**
SFk-NN/C	**2,2**	**2,5**	3,3	5,5	3,4	3,8

8.6 Badanie wpływu selekcji cech na jakość klasyfikacji

Wpływ selekcji cech na jakość klasyfikacji danych oszacowano w sposób następujący:

a) W pierwszej kolejności przeprowadzana jest sub-optymalna selekcja cech metodami opisanymi w Rozdziale 5. Powody powstawania sub-optymalnych podzbiorów cech opisane zostały w tym samym Rozdziale. Badania zostały wykonane dla wszystkich baz przedstawionych w Tabeli 8.1. Sub-optymalne zbiory cech wyznaczono po losowym wprowadzeniu do każdej z baz 25% wartości *null*.
   $\tilde{F}$-elementowe sub-optymalne zbiory cech były wyznaczane następująco:
   a) dla metod rankingowych: ReliefF [35], CSF [23], korelacji Pearsona (PC) [11], zysku informacyjnego (IG, *ang.* Information Gain) i względnego zysku informacyjnego (GR, *ang.* Gain Ratio) [25] – na podstawie wyników generowanych w pakiecie WEKA,
   b) dla indywidualnego rankingu cech – na podstawie klasyfikatora SFk-NN/C i współczynników F-Measure/G-Measure: SR(F)/SR(G),
   c) dla rankingu cech dla klas – na podstawie klasyfikatora SFk-NN/C i współczynników F-Measure/G-Measure: CR(F)/CR(G),
   d) dla metod opakowanych: selekcji w przód (FFS) i w tył (BFS) – na podstawie klasyfikatora SFk-NN/C i współczynnika ACC.

b) W kolejnym etapie, w oparciu o wygenerowane zbiory sub-optymalnych cech przeprowadzona była krzyżowa walidacja w celu ustalenia dokładności klasyfikatora. W badaniach użyto proponowanego w rozprawie klasyfikatora komitetowego SFk-NN/C. Zastosowanie tego klasyfikatora podyktowane było uzyskiwaną przez niego największą dokładnością

klasyfikacji w porównaniu z innymi, opisywanymi w pracy klasyfikatorami (patrz Tabele 8.15 – 8.19). Należy zaznaczyć, że badania przeprowadzano na wariantach baz zawierających od 5% do 35% wartości *null* z gradacją co 5%.

Dobór optymalnej liczby cech w zależności od algorytmu selekcji cech

Kolejne tabele przedstawiają wyniki ogólnej dokładności klasyfikacji baz testowych w oparciu o wstępnie wyselekcjonowane cechy. Algorytmy selekcji cech nie posiadały kryterium stopu. Testowane były zatem wszystkie możliwe do uzyskania, poszczególnymi metodami selekcji, podzbiory cech. Ostatecznie wybrany został najlepszy zestaw cech (zaznaczony pogrubiona czcionką) ustalony w oparciu o każdą z testowanych metod selekcji cech. Przez najlepszy zestaw cech należy rozumieć taki, w oparciu o który uzyskana była najwyższa ogólna dokładność klasyfikacji. Jeśli taka sama dokładność była uzyskana dla więcej niż jednego podzbioru cech, jako najlepszy, wybierany był zbiór najmniej liczny.

Tabela 8.35: Ogólna dokładność klasyfikacji (ACC) danych z bazy *WINE* w funkcji liczby wybranych cech w oparciu o metody rankingowe.

Liczba cech	Metoda selekcji cech				
	CFS	ReliefF	PC	GR	IG
1	0,697	0,590	0,584	0,584	0,697
2	0,837	0,781	0,669	0,713	0,837
3	0,876	0,882	0,747	0,876	0,843
4	0,910	0,888	0,865	0,882	0,899
5	0,927	0,916	0,893	0,893	0,916
6	**0,933**	0,904	**0,933**	**0,933**	**0,933**
7	0,916	0,916	0,916	0,927	0,916
8	0,921	0,921	0,933	0,921	0,921
9	0,916	0,916	0,916	0,927	0,916
10	0,921	0,921	0,921	0,921	0,921
11	0,933	0,916	0,921	0,933	0,916
12	0,933	**0,933**	0,916	0,933	0,933
13	0,933	0,933	0,933	0,933	0,933

Tabela 8.36: Ogólna dokładność klasyfikacji (ACC) danych z bazy *WINE* w funkcji liczby wybranych cech w oparciu o metody opakowane.

Liczba	Ranking pojedynczy		Rankingi dla klas		Przeszukiwanie	
cech	SR(F)	SR(G)	CR(F)	CR(G)	FFS	w BFS
1	0,697	0,697			0,697	0,697
2	0,747	0,837			0,837	0,837
3	0,882	0,882	0,803	0,843	0,882	0,876
4	0,882	0,882			0,899	0,910
5	0,927	0,927			0,927	0,916
6	**0,933**	**0,933**	**0,933**	**0,933**	0,933	0,927
7	0,916	0,916			**0,944**	**0,949**
8	0,921	0,933			0,944	0,949
9	0,916	0,916	0,927	0,927	0,938	0,944
10	0,921	0,921			0,944	0,944
11	0,933	0,933			0,944	0,949
12	0,933	0,933	0,933	0,933	0,933	0,949
13	0,933	0,933	0,933	0,933	0,933	0,933

Tabela 8.37: Ogólna dokładność klasyfikacji (ACC) danych z bazy *WDBC* w funkcji liczby wybranych cech w oparciu o metody rankingowe.

Liczba cech	Metoda selekcji cech				
	CFS	ReliefF	PC	GR	IG
1	0,845	0,817	0,817	0,845	0,845
2	0,884	0,794	0,893	0,884	0,888
3	0,914	0,861	0,909	0,902	0,905
4	0,907	0,905	0,924	0,902	0,909
5	0,916	0,916	0,926	0,919	0,914
6	0,926	0,910	0,933	0,926	0,923
7	0,931	0,921	0,923	0,931	0,923
8	0,926	0,926	0,924	0,926	0,924
9	0,931	0,926	0,930	**0,933**	0,924
10	0,937	0,926	0,928	0,928	0,928
11	0,937	0,935	0,931	0,926	0,926
12	0,931	0,940	0,928	0,928	0,928
13	0,926	0,940	0,921	0,921	0,930
14	0,931	0,942	0,923	0,926	0,931
15	0,937	0,940	0,923	0,923	0,923
16	0,938	0,938	0,921	0,923	0,919
17	0,942	0,942	0,928	0,926	0,917
18	**0,946**	0,946	0,933	0,924	0,919
19	0,942	0,940	0,937	0,926	0,933
20	0,942	0,937	0,938	0,926	0,933
21	0,937	0,937	**0,940**	0,926	0,933
22	0,933	0,944	0,933	0,933	**0,937**
23	0,931	**0,947**	0,931	0,933	0,933
24	0,928	0,937	0,924	0,928	0,928
25	0,924	0,937	0,924	0,924	0,924
26	0,921	0,940	0,924	0,921	0,921
27	0,928	0,937	0,924	0,928	0,928
28	0,930	0,933	0,926	0,930	0,930
29	0,928	0,933	0,933	0,930	0,930
30	0,931	0,931	0,931	0,931	0,931

Tabela 8.38: Ogólna dokładność klasyfikacji (ACC) danych z bazy *WDBC* w funkcji liczby wybranych cech w oparciu o metody opakowane.

Liczba	Ranking pojedynczy		Rankingi dla klas		Przeszukiwanie	
cech	SR(F)	SR(G)	CR(F)	CR(G)	FFS	w BFS
1	0,845	0,845			0,845	0,845
2	0,889	0,889	0,889	0,889	0,903	0,884
3	0,896	0,896			0,917	0,912
4	0,912	0,912	0,912	0,912	0,923	0,914
5	0,919	0,917			0,926	0,921
6	0,926	0,926	0,926	0,926	0,931	0,930
7	**0,938**	**0,938**			0,938	0,937
8	0,935	0,935	0,935	0,935	0,942	0,940
9	0,926	0,926			0,942	0,944
10	0,923	0,923	0,923	0,923	0,946	0,942
11	0,926	0,926			0,947	0,947
12	0,928	0,928	0,923	0,928	0,946	0,949
13	0,930	0,930			0,947	0,947
14	0,926	0,926	0,926	0,926	0,949	0,947
15	0,923	0,923			0,949	0,953
16	0,924	0,924	0,924	0,924	0,949	0,958
17	0,930	0,930			0,949	0,956
18	0,937	0,937	**0,937**	0,937	0,951	0,960
19	0,935	0,935			0,954	0,961
20	0,938	0,938	0,926	**0,938**	0,953	0,961
21	0,937	0,937			0,953	**0,963**
22	0,937	0,937	0,928	0,937	0,951	0,961
23	0,933	0,933			0,953	0,956
24	0,928	0,928	0,928	0,928	0,953	0,956
25	0,924	0,928			0,953	0,953
26	0,924	0,924	0,924	0,924	**0,956**	0,951
27	0,931	0,931			0,949	0,949
28	0,928	0,928	0,928	0,928	0,942	0,946
29	0,928	0,928			0,937	0,940
30	0,931	0,931	0,931	0,931	0,931	0,931

Tabela 8.39: Ogólna dokładność klasyfikacji (ACC) danych z bazy *MESS* w funkcji liczby wybranych cech w oparciu o metody rankingowe.

Liczba cech	Metoda selekcji cech				
	CFS	ReliefF	PC	GR	IG
1	0,555	0,533	0,533	0,555	0,533
2	0,593	0,594	0,594	0,575	0,594
3	0,612	0,590	0,599	0,583	0,608
4	0,626	0,613	0,628	0,605	0,626
5	0,616	0,626	0,622	0,611	0,616
6	0,617	0,628	0,628	0,616	0,617
7	**0,631**	0,632	0,625	0,629	0,631
8	0,613	**0,637**	0,621	0,627	0,633
9	0,617	0,620	0,629	0,620	0,617
10	0,626	0,633	0,638	0,626	0,626
11	0,629	0,633	0,637	0,636	0,629
12	0,627	0,616	0,640	0,627	0,627
13	0,627	0,613	0,638	0,627	0,627
14	0,620	0,606	0,640	0,632	0,632
15	0,613	0,606	0,639	0,630	0,630
16	0,622	0,602	**0,641**	**0,639**	**0,639**
17	0,622	0,606	0,638	0,639	0,639
18	0,627	0,620	0,629	0,639	0,639
19	0,622	0,622	0,622	0,622	0,622

Tabela 8.40: Ogólna dokładność klasyfikacji (ACC) danych z bazy *MESS* w funkcji liczby wybranych cech w oparciu o metody opakowane.

Liczba	Ranking pojedynczy		Rankingi dla klas		Przeszukiwanie	
cech	SR(F)	SR(G)	CR(F)	CR(G)	FFS	w BFS
1	0,536	0,605			0,605	0,605
2	0,606	0,613	0,581	0,613	0,613	0,611
3	0,585	0,618			0,620	0,611
4	0,596	0,631	0,618	0,610	0,623	0,620
5	0,613	0,625			0,629	0,627
6	0,626	0,631	0,629	0,628	0,630	0,633
7	0,630	0,626			0,633	0,643
8	0,627	0,628	0,630	0,633	0,643	0,655
9	0,632	0,634			0,648	**0,658**
10	**0,640**	0,637	0,634	**0,641**	0,649	0,646
11	0,640	**0,640**			**0,653**	0,648
12	0,627	0,627	**0,640**	0,623	0,643	0,642
13	0,629	0,625			0,651	0,644
14	0,629	0,633	0,629	0,633	0,653	0,645
15	0,626	0,627			0,648	0,649
16	0,637	0,633	0,626	0,633	0,650	0,650
17	0,636	0,638			0,641	0,641
18	0,623	0,622	0,639	0,622	0,639	0,639
19	0,622	0,622	0,622	0,622	0,622	0,622

Tabela 8.41: Ogólna dokładność klasyfikacji (ACC) danych z bazy *CTG* w funkcji liczby wybranych cech w oparciu o metody rankingowe.

Liczba cech	Metoda selekcji cech				
	CFS	ReliefF	PC	GR	IG
1	0,766	0,762	0,766	0,803	0,779
2	0,775	0,690	0,767	0,805	0,782
3	0,801	0,754	0,768	0,729	0,798
4	0,824	0,797	0,813	0,806	0,824
5	0,840	0,818	0,827	0,838	0,823
6	0,846	0,825	0,837	0,849	0,829
7	0,850	0,849	0,843	0,849	0,842
8	0,856	0,850	0,852	0,844	0,842
9	0,858	0,865	0,854	0,850	0,850
10	0,857	0,867	0,854	0,866	0,850
11	0,854	0,870	0,844	0,861	0,855
12	0,857	0,870	0,853	0,858	0,857
13	0,857	0,870	0,858	0,860	0,856
14	0,860	0,868	0,860	0,862	0,860
15	0,863	0,865	0,863	0,859	0,863
16	0,864	0,863	0,864	0,862	0,860
17	0,861	0,866	0,861	0,860	0,861
18	0,862	**0,871**	0,860	0,862	0,865
19	**0,867**	0,860	0,861	0,861	0,860
20	0,864	0,864	0,864	0,864	0,864
21	0,866	0,866	**0,866**	**0,866**	**0,866**

Tabela 8.42: Ogólna dokładność klasyfikacji (ACC) danych z bazy *CTG* w funkcji liczby wybranych cech w oparciu o metody opakowane.

Liczba	Ranking pojedynczy		Rankingi dla klas		Przeszukiwanie	
cech	SR(F)	SR(G)	CR(F)	CR(G)	FFS	w BFS
1	0,803	0,803			0,803	0,803
2	0,718	0,805			0,820	0,807
3	0,772	0,812	0,775	0,828	0,828	0,814
4	0,815	0,806			0,833	0,823
5	0,822	0,833			0,836	0,839
6	0,839	0,849	0,829	0,831	0,848	0,853
7	0,848	0,849			0,857	0,857
8	0,857	0,857			0,860	0,865
9	0,863	0,862	0,857	0,854	0,865	0,871
10	0,858	0,856			0,869	0,874
11	0,854	0,860			0,874	**0,881**
12	0,856	0,860	0,862	0,856	**0,881**	0,881
13	0,860	0,860			0,881	0,881
14	0,859	0,857			0,878	0,878
15	0,863	0,861	**0,869**	0,861	0,878	0,878
16	0,864	0,859			0,877	0,877
17	0,867	0,861			0,878	0,878
18	**0,868**	0,865	0,859	0,864	0,878	0,878
19	0,867	**0,867**			0,876	0,872
20	0,864	0,864			0,870	0,870
21	0,866	0,866	0,866	**0,866**	0,866	0,866

Tabela 8.43: Ogólna dokładność klasyfikacji (ACC) danych z bazy *IONO* w funkcji liczby wybranych cech w oparciu o metody rankingowe.

Liczba cech	Metoda selekcji cech				
	CFS	ReliefF	PC	GR	IG
1	0,715	0,707	0,715	0,715	0,729
2	0,781	0,792	0,755	0,798	0,809
3	0,852	0,806	0,852	0,786	0,835
4	0,866	0,858	0,829	0,801	0,852
5	0,869	0,858	0,826	0,821	0,866
6	0,866	0,863	0,852	0,855	0,863
7	0,877	0,863	0,855	0,875	0,877
8	0,880	0,866	0,866	0,877	0,883
9	0,895	0,863	0,872	0,880	0,877
10	0,889	0,866	0,869	0,883	0,863
11	**0,900**	0,883	0,869	0,883	0,866
12	0,897	**0,895**	0,877	0,889	0,866
13	0,900	0,892	0,877	**0,892**	0,863
14	0,900	0,886	0,883	0,886	0,866
15	0,900	0,892	**0,892**	0,889	0,863
16	0,897	0,889	0,886	0,883	0,863
17	0,889	0,883	0,886	0,883	0,869
18	0,886	0,872	0,883	0,886	0,877
19	0,886	0,877	0,883	0,880	0,875
20	0,886	0,880	0,877	0,886	0,877
21	0,880	0,872	0,877	0,886	**0,883**
22	0,875	0,869	0,889	0,886	0,880
23	0,875	0,872	0,889	0,886	0,872
24	0,866	0,875	0,875	0,886	0,875
25	0,875	0,875	0,877	0,889	0,877
26	0,872	0,869	0,886	0,886	0,880
27	0,875	0,866	0,880	0,877	0,880
28	0,872	0,869	0,875	0,872	0,875
29	0,872	0,869	0,872	0,866	0,875
30	0,863	0,869	0,872	0,863	0,877
31	0,863	0,869	0,866	0,863	0,869
32	0,863	0,869	0,858	0,863	0,863
33	0,863	0,863	0,863	0,863	0,863
34	0,863	0,863	0,863	0,863	0,863

Tabela 8.44: Ogólna dokładność klasyfikacji (ACC) danych z bazy *IONO* w funkcji liczby wybranych cech w oparciu o metody opakowane.

Liczba	Ranking pojedynczy		Rankingi dla klas		Przeszukiwanie	
cech	SR(F)	SR(G)	CR(F)	CR(G)	FFS	w BFS
1	0,729	0,641			0,729	0,715
2	0,818	0,801	0,818	0,801	0,843	0,832
3	0,846	0,872			0,886	0,852
4	0,843	0,872	0,840	0,872	0,900	0,869
5	0,855	0,869			0,903	0,886
6	0,872	0,866	0,866	0,866	0,909	0,889
7	0,872	0,860			0,920	0,897
8	0,872	0,872	0,866	0,872	**0,923**	0,895
9	0,872	0,866			0,923	0,895
10	0,866	0,860	0,877	0,860	0,923	0,900
11	0,877	0,872			0,920	0,903
12	0,869	0,872	0,880	0,872	0,917	0,895
13	0,866	0,877			0,920	0,903
14	0,866	**0,892**	0,889	**0,892**	0,920	0,906
15	0,875	0,880			0,917	0,909
16	0,880	0,875	0,880	0,875	0,917	0,909
17	0,880	0,866			0,912	0,906
18	0,886	0,869	0,886	0,869	0,909	0,909
19	0,883	0,880			0,909	**0,912**
20	**0,892**	0,880	**0,892**	0,880	0,906	0,909
21	0,886	0,883			0,903	0,906
22	0,880	0,877	0,875	0,877	0,903	0,906
23	0,877	0,872			0,900	0,897
24	0,877	0,872	0,869	0,872	0,900	0,897
25	0,872	0,872			0,897	0,897
26	0,872	0,872	0,877	0,872	0,892	0,892
27	0,872	0,872			0,883	0,889
28	0,869	0,875	0,869	0,875	0,883	0,889
29	0,872	0,866			0,880	0,889
30	0,872	0,872	0,872	0,872	0,883	0,886
31	0,866	0,872			0,883	0,883
32	0,863	0,863	0,863	0,863	0,875	0,877
33	0,863	0,863			0,875	0,875
34	0,863	0,863	0,863	0,863	0,863	0,863

Tabela 8.45: Ogólna dokładność klasyfikacji (ACC) danych z bazy *CRYO* w funkcji liczby wybranych cech w oparciu o metody rankingowe.

Liczba cech	Metoda selekcji cech				
	CFS	ReliefF	PC	GR	IG
1	0,689	0,700	0,700	0,689	0,700
2	0,789	0,789	0,789	0,789	0,789
3	0,767	0,767	0,767	0,767	0,767
4	0,778	**0,822**	**0,822**	**0,822**	**0,822**
5	0,800	0,822	0,822	0,822	0,822
6	**0,811**	0,811	0,811	0,811	0,811

Tabela 8.46: Ogólna dokładność klasyfikacji (ACC) danych z bazy *CRYO* w funkcji liczby wybranych cech w oparciu o metody opakowane.

Liczba cech	Ranking pojedynczy		Rankingi dla klas		Przeszukiwanie	
	SR(F)	SR(G)	CR(F)	CR(G)	FFS	w BFS
1	0,700	0,700			0,700	0,700
2	0,700	0,700	0,700	0,700	0,789	0,789
3	0,767	0,767			0,800	0,800
4	**0,811**	**0,811**	**0,811**	**0,811**	0,811	**0,822**
5	0,800	0,800			**0,822**	0,822
6	0,811	0,811	0,811	0,811	0,811	0,811

Analiza zawartości Tabel 8.35 – 8.46 wskazuje, że algorytmy selekcji cech były najmniej efektywne w odniesieniu do bazy *IONO*, gdzie dla 4 z 11 algorytmów wyznaczony podzbiór cech był równy pełnemu zbiorowi cech. Dla wszystkich pozostałych baz danych zastosowane algorytmy były w stanie wyselekcjonować podzbiór cech, dla których uzyskano lepszą lub nie gorszą jakość klasyfikacji w porównaniu ze pełnym zbiorem cech.

Zaletą metod rankingowych (oraz metod opakowanych wykorzystujących pojedyncze rankingi) jest szybkość ich działania. Wyznaczenie optymalnego zbioru cech bez użycia rankingu wymagałoby sprawdzenia 2^F zestawów cech. W przypadku pojedynczego rankingu sub-optymalny wynik zostanie uzyskany po sprawdzeniu nie więcej niż F zestawów cech. Dla oddzielnych rankingów poszczególnych klas sprawdzeń będzie nie więcej niż $^F/_L + 1$, gdzie F jest liczbą cech, a L jest liczbą klas. Metody przeszukiwania w przód i w tył, są bardziej kosztowne obliczeniowo, na przykład dla bazy *IONO* wymagane było sprawdzenie 595 zestawów cech w stosunku do 34 w metodach rankingowych. Zatem metody przeszukiwania w przód/w tył nie sprawdzą się w zastosowaniach, gdzie zbiór referencyjny jest często uzupełniany, a selekcja cech musi być

powtórzona jako element klasyfikacji. Dla dużych zbiorów danych może to być warunek krytyczny ze względu na czas obliczeń.

Badanie wpływu liczby brakujących wartości cech na jakość klasyfikacji po wstępnej selekcji cech

Podstawowym założeniem rozprawy było opracowanie metody, dla której można uzyskać dobrą jakość klasyfikacji na podstawie niekompletnych danych. Z tego względu dokonano ponownej oceny wpływu liczby brakujących wartości cech na dokładność klasyfikacji według scenariusza „zdegradowany testowany zdegradowanym", a wyniki zostały przedstawione w poniższych Tabelach. Scenariusz badań jest opisany w Rozdziale 8.4. **Badania te stanowią potwierdzenie realizacji piątego celu pracy.**

Tabela 8.47: Ogólna dokładność klasyfikacji (ACC) danych z bazy *WINE* według scenariusza „zdegradowany testowany zdegradowanym" w oparciu o wstępnie wyselekcjonowane cechy.

Metoda		Procentowa liczba brakujących cech w wektorze **x**									
selekcji	$\tilde{F}$	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$	s^2	R
BRAK	13	0,940	0,943	0,926	0,922	**0,916**	**0,907**	**0,890**	0,921	0,000	0,053
CFS	6	0,937	0,932	0,922	0,921	0,896	0,892	0,880	0,911	0,000	0,057
RelieF	12	0,942	0,940	0,929	0,920	0,915	0,906	0,893	0,921	0,000	0,049
PC	6	0,937	0,932	0,922	0,921	0,896	0,892	0,880	0,911	0,000	0,057
GR	6	0,937	0,932	0,922	0,921	0,896	0,892	0,880	0,911	0,000	0,057
IG	6	0,937	0,932	0,922	0,921	0,896	0,892	0,880	0,911	0,000	0,057
SR(F)	6	0,937	0,932	0,922	0,921	0,896	0,892	0,880	0,911	0,000	0,057
SR(G)	6	0,937	0,932	0,922	0,921	0,896	0,892	0,880	0,911	0,000	0,057
CR(F)	6	0,937	0,932	0,922	0,921	0,896	0,892	0,880	0,911	0,000	0,057
CR(G)	6	0,937	0,932	0,922	0,921	0,896	0,892	0,880	0,911	0,000	0,057
FFS	7	**0,956**	**0,947**	**0,934**	**0,929**	0,907	0,902	0,887	**0,923**	0,001	0,070
BFS	7	0,948	0,947	0,933	0,919	0,904	0,893	0,869	0,916	0,001	0,079

Tabela 8.48: Ogólna dokładność klasyfikacji (ACC) danych z bazy *WDBC* według scenariusza „zdegradowany testowany zdegradowanym" w oparciu o wstępnie wyselekcjonowane cechy.

Metoda		Procentowa liczba brakujących cech w wektorze $\mathbf{x}$									
selekcji	$\tilde{F}$	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$	s^2	R
BRAK	30	0,932	0,931	0,925	0,923	0,923	0,921	0,910	0,924	0,000	0,022
CFS	18	0,938	0,937	0,938	0,935	0,931	0,933	**0,931**	**0,935**	0,000	0,007
RelieF	23	0,941	0,938	0,937	0,934	0,932	0,931	0,925	0,934	0,000	0,016
PC	21	0,941	0,938	0,936	0,933	0,931	0,934	0,925	0,934	0,000	0,016
GR	9	0,932	0,931	0,929	0,926	0,925	0,925	0,920	0,927	0,000	0,012
IG	21	0,935	0,931	0,931	0,925	0,924	0,924	0,919	0,927	0,000	0,016
SR(F)	7	0,932	0,930	0,931	0,926	0,925	0,927	0,922	0,928	0,000	0,009
SR(G)	7	0,932	0,930	0,931	0,926	0,925	0,927	0,922	0,928	0,000	0,009
CR(F)	18	0,940	0,938	0,937	0,931	0,931	**0,935**	0,928	0,934	0,000	0,011
CR(G)	20	0,942	0,940	0,937	0,931	0,930	0,933	0,927	0,934	0,000	0,015
FFS	26	0,944	0,944	**0,940**	0,934	**0,933**	0,930	0,922	0,935	0,000	0,023
BFS	21	**0,946**	**0,945**	**0,940**	**0,935**	0,930	0,929	0,920	0,935	0,000	0,025

Tabela 8.49: Ogólna dokładność klasyfikacji (ACC) danych z bazy *MESS* według scenariusza „zdegradowany testowany zdegradowanym" w oparciu o wstępnie wyselekcjonowane cechy.

Metoda		Procentowa liczba brakujących cech w wektorze $\mathbf{x}$									
selekcji	$\tilde{F}$	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$	s^2	R
BRAK	19	0,621	0,611	0,616	0,611	0,614	0,602	0,597	0,610	0,000	0,024
CFS	7	0,624	0,624	0,619	0,617	0,614	0,613	0,607	0,617	0,000	0,017
RelieF	8	0,617	0,616	0,613	0,613	0,612	0,604	0,602	0,611	0,000	0,015
PC	16	0,629	0,621	0,626	0,617	0,617	0,612	0,604	0,618	0,000	0,025
GR	16	0,621	0,616	0,618	0,616	0,619	0,610	0,601	0,614	0,000	0,021
IG	16	0,621	0,616	0,618	0,616	0,619	0,610	0,601	0,614	0,000	0,021
SR(F)	10	0,626	0,626	0,623	0,623	0,615	0,613	**0,610**	0,619	0,000	0,016
SR(G)	11	**0,635**	0,627	0,624	0,621	0,618	0,616	0,609	0,622	0,000	0,026
CR(F)	12	0,634	0,627	0,625	0,622	0,618	**0,617**	0,609	0,622	0,000	0,026
CR(G)	10	0,633	0,626	0,623	0,620	0,616	0,613	0,607	0,620	0,000	0,025
FFS	11	0,630	0,621	0,622	0,617	0,619	0,609	0,610	0,618	0,000	0,022
BFS	9	0,634	**0,629**	**0,628**	**0,623**	**0,621**	0,611	0,608	**0,622**	0,000	0,026

Tabela 8.50: Ogólna dokładność klasyfikacji (ACC) danych z bazy *CTG* według scenariusza „zdegradowany testowany zdegradowanym" w oparciu o wstępnie wyselekcjonowane cechy.

Metoda		Procentowa liczba brakujących cech w wektorze **x**									
selekcji	$\tilde{F}$	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$	s^2	R
BRAK	21	0,868	0,868	0,870	0,865	0,864	0,862	0,859	0,865	0,000	0,011
CFS	19	0,869	0,868	0,869	0,864	0,864	0,863	0,856	0,865	0,000	0,013
RelieF	18	0,873	0,873	0,872	0,867	0,866	0,862	0,858	0,867	0,000	0,015
PC	21	0,868	0,868	0,870	0,865	0,864	0,862	0,859	0,865	0,000	0,011
GR	21	0,868	0,868	0,870	0,865	0,864	0,862	0,859	0,865	0,000	0,011
IG	21	0,868	0,868	0,870	0,865	0,864	0,862	0,859	0,865	0,000	0,011
SR(F)	18	0,869	0,869	0,870	0,864	0,864	0,863	0,858	0,865	0,000	0,012
SR(G)	19	0,869	0,868	0,869	0,864	0,864	0,863	0,856	0,865	0,000	0,013
CR(F)	15	0,875	0,872	0,873	0,867	0,866	0,862	0,857	0,867	0,000	0,017
CR(G)	21	0,868	0,868	0,870	0,865	0,864	0,862	0,859	0,865	0,000	0,011
FFS	12	0,879	**0,879**	**0,877**	**0,872**	**0,873**	**0,865**	**0,863**	**0,873**	0,000	0,016
BFS	11	**0,880**	0,878	**0,877**	0,870	0,871	0,864	0,860	0,871	0,000	0,020

Tabela 8.51: Ogólna dokładność klasyfikacji (ACC) danych z bazy *IONO* według scenariusza „zdegradowany testowany zdegradowanym" w oparciu o wstępnie wyselekcjonowane cechy.

Metoda		Procentowa liczba brakujących cech w wektorze **x**									
selekcji	$\tilde{F}$	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$	s^2	R
BRAK	34	0,874	0,873	0,872	0,877	0,870	0,872	0,873	0,873	0,000	0,007
CFS	11	0,901	0,900	0,896	**0,891**	**0,885**	**0,885**	0,872	**0,890**	0,000	0,029
RelieF	12	0,897	0,888	0,892	0,887	**0,885**	0,877	0,870	0,885	0,000	0,027
PC	15	0,887	0,886	0,884	0,885	0,877	0,879	0,877	0,882	0,000	0,010
GR	13	0,885	0,884	0,887	0,881	0,876	0,875	0,872	0,880	0,000	0,015
IG	14	0,871	0,872	0,875	0,879	0,873	0,874	0,870	0,873	0,000	0,009
SR(F)	20	0,883	0,878	0,878	0,879	0,880	0,880	0,875	0,879	0,000	0,008
SR(G)	14	0,884	0,881	0,885	0,881	0,869	0,873	0,866	0,877	0,000	0,020
CR(F)	20	0,883	0,878	0,878	0,879	0,880	0,880	0,875	0,879	0,000	0,008
CR(G)	14	0,884	0,881	0,885	0,881	0,869	0,873	0,866	0,877	0,000	0,020
FFS	8	**0,906**	**0,906**	**0,898**	0,887	0,882	0,876	0,866	0,889	0,000	0,040
BFS	19	0,887	0,888	0,889	0,884	0,884	0,884	**0,878**	0,885	0,000	0,011

Tabela 8.52: Ogólna dokładność klasyfikacji (ACC) danych z bazy *CRYO* według scenariusza „zdegradowany testowany zdegradowanym" w oparciu o wstępnie wyselekcjonowane cechy.

Metoda		Wyrażona w % liczba brakujących cech w wektorze **x**									
selekcji	$\tilde{F}$	5%	10%	15%	20%	25%	30%	35%	$\overline{ACC}$	s^2	R
BRAK	6	0,838	0,830	0,841	0,824	**0,828**	0,814	0,781	0,822	0,000	0,060
CFS	6	0,838	0,830	0,841	0,824	**0,828**	0,814	0,781	0,822	0,000	0,060
RelieF	4	**0,844**	0,838	0,833	**0,831**	0,819	**0,827**	**0,794**	**0,827**	0,000	0,050
PC	4	**0,844**	0,838	0,833	**0,831**	0,819	**0,827**	**0,794**	**0,827**	0,000	0,050
GR	4	**0,844**	0,838	0,833	**0,831**	0,819	**0,827**	**0,794**	**0,827**	0,000	0,050
IG	4	**0,844**	0,838	0,833	**0,831**	0,819	**0,827**	**0,794**	**0,827**	0,000	0,050
SR(F)	4	0,681	0,574	0,580	0,576	0,571	0,565	0,562	0,587	0,002	0,119
SR(G)	4	0,821	0,816	0,829	0,820	0,810	0,799	0,792	0,812	0,000	0,037
CR(F)	4	0,821	0,816	0,829	0,820	0,810	0,799	0,792	0,812	0,000	0,037
CR(G)	4	0,821	0,816	0,829	0,820	0,810	0,799	0,792	0,812	0,000	0,037
FFS	5	0,842	**0,840**	**0,852**	0,828	0,821	0,820	0,787	**0,827**	0,000	0,066
BFS	4	**0,844**	0,838	0,833	**0,831**	0,819	**0,827**	**0,794**	**0,827**	0,000	0,050

Należy odnotować, że redukcja części cech oznaczała zmniejszenie liczby klasyfikatorów k-NN w komitetach SFk-NN(/C). Pomimo tego można zauważyć, że proponowany w rozprawie klasyfikator SFk-NN/C dla każdej metody selekcji cech i każdej bazy danych zapewnił stabilną jakości klasyfikacji w całym zakresie (5% – 35%) brakujących wartości cech.
Stanowi to potwierdzenie Tezy postawionej w Rozprawie.

8.7 Testowanie klasyfikatora SFk-NN/C na rzeczywistych danych medycznych

W tym Podrozdziale opisane zostały wyniki badań mających na celu weryfikację działania klasyfikatora na rzeczywistych danych medycznych, które w żaden sposób nie były modyfikowane (uzupełniane ani równoważone). Badania te posłużyły do określenia przydatności klasyfikatora SFk-NN/C do budowy systemu oceny stopnia włóknienia wątroby u pacjentów z wirusowym zapaleniem wątroby typu C. Badania prowadzone były w oparciu o bazę *HCV Liver Fibrosis* **i stanowią realizację siódmego celu pracy**.
Baza *HCV Liver Fibrosis* [45, 47] użyta do tych badań została utworzona przy współpracy z *Oddziałem Gastroenterologii i Hepatologii Kliniki Gastroenterologii i Hepatologii Samodzielnego Publicznego Centralnego Szpitala Klinicznego im. prof. Kornela Gibińskiego Śląskiego Uniwersytetu Medycznego w Katowicach.*

Tabela 8.53: Charakterystyka bazy danych użytej do budowy systemu.

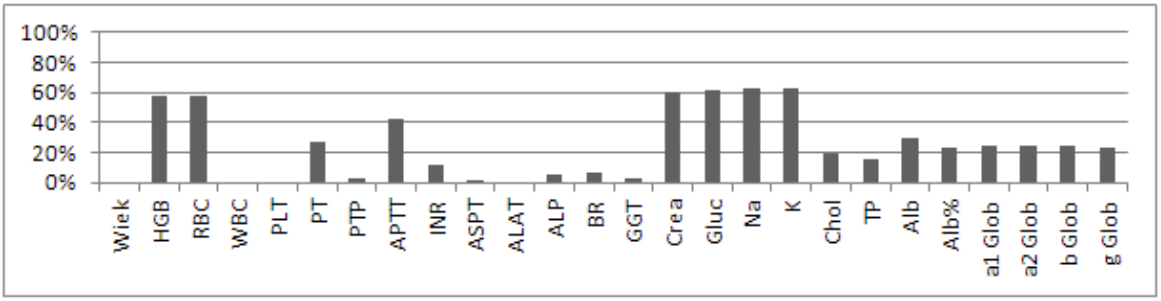
L.p.	Pełna nazwa	Skrót	Obiekty (Q)	Cechy (F)	Klasy (L)
1.	HCV Liver Fibrosis	HEPA	290	26	3

Baza ta zawiera wyłącznie anonimizowane dane pochodzące od 290 pacjentów (Tabela 8.53). Klasa mniejszościowa w tej bazie jest reprezentowana przez 32% mniej obiektów niż klasa najbardziej liczna. Średnio w bazie brakuje 25% wartości cech. W skład danych wchodzą: wiek pacjenta oraz wyniki badań krwi (serologia i hematologia, biochemia, rozmaz, elektroforeza białek). Dane te są opisane wynikiem biopsji cienko-igłowej wątroby. Choć oryginalnie wynik biopsji opisany był w 5-cio stopniowej skali *Metavir*, po konsultacji z ekspertami medycznymi, liczba klas została ograniczona do 3 (Tabela 8.54). Było to podyktowane dużą niepewnością wyniku biopsji, określaną według różnych źródeł na 25–33% [4, 49].

Tabela 8.54: Rozkład klas w bazie *HCV Liver Fibrosis*.

Klasa	METAVIR	Liczba
1	F0, F1	129
2	F2, F3	102
3	F4	59

W poszczególnych wektorach bazy *HCV Liver Fibrosis* brakuje wartości niektórych cech. W odróżnieniu od baz używanych w Rozdziale 8 rozkład brakujących wartości nie jest równomierny. Są tam obiekty (dane pacjentów) opisane kompletem cech, ale niektórych cech brakuje dla ponad 60% pacjentów. Pokazano to na Rys. 8.12. Jest to więc baza zawierająca dane trudne do analizy.



Rys. 8.12: Rozkład brakujących wartości w bazie *HCV Liver Fibrosis*.

Porównanie klasyfikatorów działających na surowych danych

Baza *HCV Liver Fibrosis*, zawiera dane uzyskane w sposób rzeczywisty, a brakujące w tej bazie dane występują z przyczyn naturalnych. Żadne cechy, ani rekordy nie były z tej bazy usuwane ani sztucznie uzupełniane. Baza była więc analizowana w takiej formie, w jakiej została pozyskana. W przypadku tej bazy jedynym scenariuszem, który można zastosować do walidacji metody jest scenariusz uczenia klasyfikatora przy pomocy niekompletnych danych i testowania również przy pomocy niekompletnych danych, czyli według wcześniejszych opisów „zdegradowany testowany zdegradowanym".

W praktyce lekarskiej korzysta się z testów diagnostycznych, których celem jest potwierdzenie lub wykluczenie choroby pacjenta. Od stopnia wiarygodności testu zależy rozpoznanie i przyszłe leczenie, dlatego do oceny ich przydatności w procesie diagnostycznym, oprócz skuteczności wykorzystuje się również inne wskaźniki, takie jak czułość i swoistość metody (patrz Tabela 3.1).
Czułość to stosunek wyników prawdziwie dodatnich do sumy prawdziwie dodatnich i fałszywie ujemnych, w odniesieniu do testów medycznych opisuje zdolność wykrywania osób rzeczywiście chorych.
Swoistość to stosunek wyników prawdziwie ujemnych do sumy prawdziwie ujemnych i fałszywie dodatnich, w odniesieniu do testów medycznych opisuje zdolność wykrywania osób rzeczywiście zdrowych.
Oddzielnie współczynniki te nie mają wartości informacyjnej, gdyż przykładowo wynik 1.0 dla czułości testu, byłby możliwy dla metody, która wskazywałaby każdego pacjenta jako chorego. Podobnie swoistość równa 1.0 możliwa byłaby do uzyskania przez metodę, która wszystkich pacjentów oznaczałaby jako zdrowych. Ponadto miary te, z definicji, przeznaczone są dla testów binarnych (zdrowy/chory), a nie problemów wielo-klasowych, z jakim mamy do czynienia w przypadku bazy danych *HCV Liver Fibrosis*. W takich przypadkach konieczne jest wyznaczenie tych parametrów oddzielnie dla każdej klasy według zasady „jedna przeciw pozostałym" i uśrednienie wyników (patrz Tabela 3.2).
Ponieważ w tej części badań analizowano zbiór rzeczywistych danych medycznych, dla zaprezentowanych w rozprawie klasyfikatorów leniwych, oprócz ogólnej dokładności klasyfikacji (ACC) wyznaczono również ich czułość (SEN) i swoistość (SPE), co zaprezentowano w Tabeli 8.55.

Tabela 8.55: Ogólna dokładność klasyfikacji, średnia czułość i swoistość danych z bazy *HCV Liver Fibrosis*.

Klasyfikator	ACC	SEN	SPE
k-NN	0,479	0,390	0,698
Wk-NN^g	0,476	0,397	0,701
Wk-NN^l	0,562	0,544	0,767
DWk-NN	0,562	0,544	0,767
EWk-NN	0,531	0,501	0,747
AFFk-NN	0,559	0,524	0,749
RSk-NN	0,455	0,373	0,681
SFk-NN	0,621	0,563	0,781
SFk-NN/C	**0,631**	**0,606**	**0,796**

Dla bazy *HCV Liver Fibrosis*, przewaga jakości klasyfikacji algorytmu SFk-NN/C jest podobna jak w przypadku baz opisanych w Rozdziale 8, gdzie klasyfikator SFk-NN/C zapewniał średnio 24% lepszą ogólną dokładność klasyfikacji w porównaniu z innymi klasyfikatorami. Proponowany w rozprawie klasyfikator SFk-NN/C uzyskał najlepszy wynik klasyfikacji zarówno pod względem dokładności klasyfikacji, jak i średniej czułości oraz swoistości.

Porównanie metod wstępnej selekcji cech

W Tabelach 8.57 – 8.56 zebrane zostały wyniki *ogólnej dokładności klasyfikacji* (ACC) dla klasyfikatora SFk-NN/C uzyskane w wyniku walidacji krzyżowej w oparciu o wstępnie wyselekcjonowane cechy. Wybór tego klasyfikatora podyktowany był uzyskiwaną przez niego największą dokładnością klasyfikacji w porównaniu z innymi, opisywanymi w pracy klasyfikatorami (patrz Podrozdział 8.7). Z tych samych powodów ranking cech tworzony był w oparciu o decyzje zwracane przez klasyfikator SFk-NN/C.

Tabela 8.56: Ogólna dokładność klasyfikacji (ACC) danych z bazy *HCV Liver Fibrosis* w funkcji liczby wybranych cech w oparciu o metody rankingowe.

Liczba cech	Metoda selekcji cech				
	CFS	ReliefF	PC	GR	IG
1	0,490	0,559	0,559	0,490	0,490
2	0,528	0,603	0,603	0,500	0,528
3	0,641	0,641	0,641	0,510	0,641
4	0,610	0,652	0,652	0,517	0,631
5	0,638	0,634	0,641	0,579	**0,679**
6	0,624	0,631	0,648	0,617	0,655
7	0,614	0,628	0,652	0,634	0,679
8	0,666	0,621	0,666	0,621	0,669
9	0,659	0,617	**0,672**	0,617	0,662
10	0,648	0,621	0,645	0,645	0,648
11	0,672	0,638	0,645	**0,659**	0,648
12	0,666	0,634	0,659	0,645	0,662
13	**0,676**	0,641	0,659	0,655	0,662
14	0,662	0,638	0,659	0,659	0,659
15	0,648	0,652	0,634	0,648	0,648
16	0,648	0,645	0,621	0,648	0,648
17	0,638	0,634	0,614	0,652	0,638
18	0,638	**0,655**	0,628	0,638	0,638
19	0,648	0,652	0,631	0,628	0,628
20	0,662	0,655	0,634	0,617	0,617
21	0,641	0,655	0,628	0,645	0,645
22	0,659	0,638	0,624	0,628	0,628
23	0,659	0,614	0,610	0,638	0,638
24	0,659	0,628	0,621	0,652	0,652
25	0,659	0,631	0,631	0,631	0,631
26	0,631	0,631	0,631	0,631	0,631

Tabela 8.57: Ogólna dokładność klasyfikacji (ACC) danych z bazy *HCV Liver Fibrosis* w funkcji liczby wybranych cech w oparciu o metody opakowane.

Liczba cech	Ranking pojedynczy SR(F)	SR(G)	Rankingi dla klas CR(F)	CR(G)	Przeszukiwanie FFS	w BFS
1	0,559	0,493			0,559	0,559
2	0,583	0,583			0,621	0,621
3	0,597	0,597	0,641	0,641	0,641	0,641
4	0,572	0,572			0,672	0,672
5	0,621	0,621			0,703	0,703
6	0,662	0,662	0,672	0,672	0,703	0,700
7	0,662	0,662			0,710	0,703
8	0,655	0,672			0,714	0,707
9	0,703	0,693	**0,703**	**0,728**	0,717	0,717
10	0,703	0,710			0,721	0,728
11	**0,717**	0,697			0,731	**0,738**
12	0,717	**0,721**	0,672	0,714	**0,734**	0,734
13	0,714	0,714			0,731	0,728
14	0,683	0,683			0,731	0,717
15	0,686	0,686	0,697	0,700	0,731	0,714
16	0,659	0,700			0,728	0,703
17	0,645	0,666			0,728	0,703
18	0,645	0,641	0,679	0,672	0,714	0,714
19	0,652	0,652			0,710	0,703
20	0,652	0,659			0,700	0,697
21	0,666	0,666	0,652	0,655	0,683	0,683
22	0,634	0,634			0,676	0,676
23	0,652	0,652			0,666	0,672
24	0,628	0,628	0,631	0,655	0,648	0,659
25	0,631	0,631			0,628	0,659
26	0,631	0,631	0,631	0,631	0,631	0,631

Dla opisywanej bazy danych *HCV Liver Fibrosis* najlepszą dokładność klasyfikacji (73,8%) udało się uzyskać dla zestawu 11 cech (spośród 26 możliwych) wyselekcjonowanych z wykorzystaniem opakowanej metody selekcji cech z przeszukiwaniem w tył. Można zauważyć, że dla klasyfikatora SFk-NN/C wstępna selekcja zawsze powodowała poprawę ogólnej dokładności klasyfikacji w porównaniu do klasyfikacji bez selekcji cech. Uzyskane wyniki klasyfikacji danych medycznych związanych z włóknieniem wątroby należ uznać za dobre, biorąc pod uwagę fakt, że uczenie klasyfikatora przebiegało na podstawie wskazań eksperta hepatologa. Z konsultacji

z ekspertami medycznymi wynika, że ich trafność diagnozy jest na podobnym poziomie. Jakość tej diagnozy można poprawić dodatkowymi procedurami medycznymi, które w tej rozprawie nie były brane pod uwagę.

Dla podzbioru cech, który umożliwił uzyskanie najlepszego pod względem dokładności klasyfikacji z użyciem klasyfikatora SFk-NN/C, powtórzone zostało badanie polegające na wyznaczeniu ogólnej dokładności klasyfikacji, czułości oraz specyficzności wszystkich testowanych dotychczas klasyfikatorów leniwych. Wyniki zostały przedstawione w Tabeli 8.58.

Tabela 8.58: Ogólna dokładność klasyfikacji, średnia czułość i swoistość danych z bazy *HCV Liver Fibrosis*, z wstępnie wyselekcjonowanymi cechami.

Klasyfikator	ACC	SEN	SPE
k-NN	0,517	0,449	0,722
Wk-NN^g	0,521	0,457	0,728
Wk-NN^l	0,566	0,550	0,769
DWk-NN	0,566	0,550	0,769
EWk-NN	0,562	0,526	0,766
AFFk-NN	0,600	0,565	0,778
RSk-NN	0,531	0,473	0,730
SFk-NN	0,700	0,643	0,831
SFk-NN/C	**0,738**	**0,719**	**0,859**

Na powyższym przykładzie można zaobserwować poprawę wszystkich zmierzonych wskaźników dla wszystkich klasyfikatorów jaka zaszła w stosunku do klasyfikacji oryginalnego zbioru danych bez wstępnej selekcji cech (patrz Tabela 8.55).

Rozdział 9

Podsumowanie

Za najważniejsze z osiągnięć należy uznać utrzymanie dobrej ogólnej jakości klasyfikacji proponowanego klasyfikatora SFk-NN/C w szerokim zakresie liczby brakujących danych. W większości klasyfikowanych baz klasyfikator SFk-NN/C zapewniał najmniejszą zmienność wyniku klasyfikacji, **co stanowi potwierdzenie tezy pracy**.

Analiza działania klasyfikatora k-NN przedstawionego w Rozdziale 6 pozwala stwierdzić, iż występowanie niepełnych wektorów (zawierających wartości *null*), zaburza jego działanie. W Podrozdziałach 6.2 oraz 7.4 przedstawiony został mechanizm powstawania niepełnych wektorów referencyjnych oraz klasyfikowanych dla podprzestrzeni cech o różnych rozmiarach. Pozwala to zaobserwować w jaki sposób liczba niepełnych wektorów zależy od rozmiaru podprzestrzeni cech na jakiej działają klasyfikatory składowe komitetu klasyfikatorów. Zależność ta została potwierdzona eksperymentalnie w Podrozdziałach 8.1 oraz 8.3. W Podrozdziale 8.4 opisano wyniki badań prowadzonych na rzeczywistych danych, ze sztucznie wprowadzonymi wartościami *null*. **Badania te stanowią realizację pierwszego celu pracy**.

W oparciu o wnioski płynące z działania klasyfikatora pracującego na niepełnych wektorach danych, w Podrozdziale 7.2 zawarty został opis proponowanej struktury komitetu klasyfikatorów, **co stanowi realizację drugiego celu pracy**.

W Podrozdziale 8.2 przedstawione zostały wyniki badań, mające na celu określenie przydatności innych klasyfikatorów do budowy komitetu klasyfikatorów o wcześniej ustalonej strukturze. Potwierdzają one, że jedynym klasyfikatorem czerpiącym korzyść z zaproponowanej struktury komitetu jest klasyfikator k-NN. **Badanie te stanowiły realizację trzeciego celu pracy**.

Przydatność klasyfikatora k-NN do budowy komitetu klasyfikatorów, odpornego na wartości *null* w klasyfikowanych danych, wykazano w Podrozdziale 8.2. Badania wykonano na podstawie głosowania decyzji klasyfikatorów składowych komitetu. Klasyfikator SFk-NN/C, opisany Podrozdziale 7.2, wykorzystuje uśrednioną wartość wsparcia dla decyzji pochodzących od klasyfikatorów składowych, czyli tzw. uśrednianie bayesowskie. W Podrozdziale 7.3 przedstawiona została modyfikacja funkcji wsparcia komitetu klasyfikatorów, uwzględniająca liczność obiektów zbioru referencyjnego należących do poszczególnych klas. Proponowana w rozprawie modyfi-

kacja poprawiła skuteczności klasyfikacji obiektów mało licznie reprezentowanych w zbiorze referencyjnym. **Zmodyfikowana funkcja wsparcia (7.9) jest realizacją czwartego celu pracy**.

Podrozdział 7.5 zawiera opis tzw. opakowanego algorytmu selekcji cech, wykorzystującego klasyfikator SFk-NN. Skuteczność tego i innych algorytmów selekcji cech wymienionych w Rozdziale 5 została zbadana metodą eksperymentalną, a wyniki zostały przedstawione w Podrozdziale 8.6. **Badania te stanowią realizację piątego celu pracy**.

Badania opisane w Podrozdziałach 8.4 i 8.6 zostały powtórzone na danych zawierających wyniki badań pacjentów zakażonych wirusem HCV-C. Baza ta, podobnie jak pozostałe, zawierała wyłącznie dane rzeczywiste i niezrównoważone, ale w odróżnieniu od wcześniej używanych baz benchmarkowych, wektory danych opisujące obiekty były niepełne, tj. zawierały wartości *null*. Niezdefiniowane wartości w tej bazie wynikały z przyczyn naturalnych, nie były sztucznie wprowadzane jak w pozostałych bazach. Wyniki badań przedstawione w Podrozdziale 8.7 potwierdziły przydatność proponowanego klasyfikatora do budowy systemu wspomagania decyzji medycznych, **co stanowiło realizację szóstego celu pracy**.

9.1 Zastosowania

Opisywane w rozprawie klasyfikatory SFk-NN(/C) przeznaczone są do klasyfikacji wielowymiarowych danych typu rzeczywistego (wielowartościowych), gdzie zarówno dane referencyjne jak i dane klasyfikowane posiadają niezdefiniowane wartości (*null*). Klasyfikator ten wykazuje odporność na niezrównoważenie klas w zbiorze referencyjnym oraz pozwala na utrzymanie dobrej jakości klasyfikacji, w szerokim zakresie degradacji danych w zbiorze uczącym oraz klasyfikowanym. Proponowany klasyfikator nie wymaga wstępnego przygotowania danych, a w szczególności uzupełniania brakujących wartości, czy równoważenia klas w zbiorze referencyjnym. Te właściwości klasyfikatora predestynują go do klasyfikacji danych medycznych. Klasyfikator tego typu może być częścią systemu wspomagania decyzji medycznych, systemu monitorowania stopnia zaawansowania chorób przewlekłych lub automatycznego systemu badań przesiewowych.

9.2 Dalsze prace

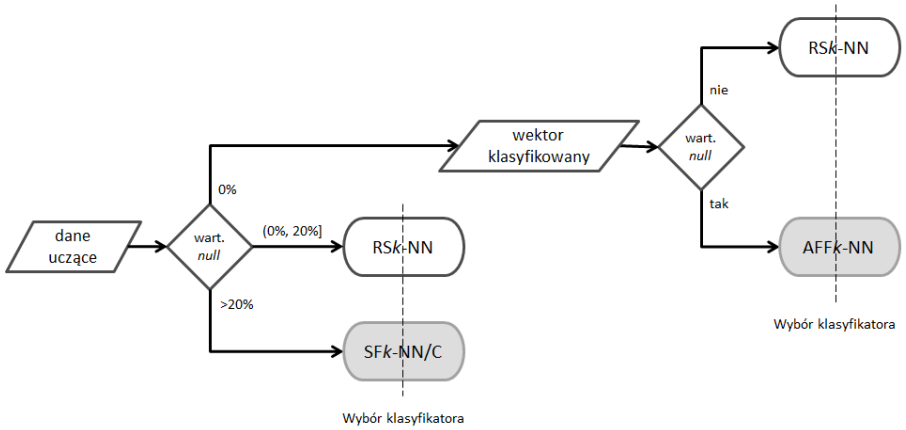
Jak wynika z badań eksperymentalnych, możliwe jest utrzymanie jakości klasyfikacji proponowanego w Rozprawie klasyfikatora komitetowego SFk-NN/C w szerokim zakresie występowania brakujących wartości (oznaczanych przez przypisanie im wartości *null*) w zbiorze danych. Jednak zależnie od miejsca występowania (wektory uczące czy dane klasyfikowane) jak i liczby

brakujących danych, najlepsze wyniki klasyfikacji możliwe są do uzyskania przy użyciu jednego z przedstawionych w pracy klasyfikatorów leniwych. Z reguły charakter danych referencyjnych jak i klasyfikowanych są z góry znane. Aby uzyskać lepszą jakość klasyfikacji, można zatem dobrać odpowiedni do danych klasyfikator.

Jeśli dane uczące są kompletne, a w danych klasyfikowanych mogą występować zarówno wektory pełne jak i niepełne, możliwe jest stworzenie klasyfikatora hybrydowego. Klasyfikator taki, zależnie od podanego wektora, klasyfikowałby go z zastosowaniem algorytmu RSk-NN dla danych pełnych, lub AFFk-NN jeśli w wektorze klasyfikowanym wystąpią wartości *null*. Z uwagi na użycie klasyfikatorów leniwych nie pociąga to za sobą konieczności utrzymywania oddzielnych kopii danych referencyjnych, ani tworzenia dodatkowych modeli decyzyjnych.

Jeśli dane uczące są niepełne to w zależności od liczby wartości *null* w zbiorze uczącym może być użyty klasyfikator RSk-NN, jeśli w zbiorze uczącym brakuje nie więcej niż 20% wartości, lub klasyfikator SFk-NN/C, jeśli ta liczba jest większa.

W przypadku klasyfikacji danych, gdzie wartości *null* występują zarówno w zbiorze referencyjnym, jak i klasyfikowanym możliwa jest fuzja obydwu rozwiązań (Rys. 9.1).



Rys. 9.1: Schemat wyboru algorytmu klasyfikacji w klasyfikatorze hybrydowym.

Skuteczność przedstawionego rozwiązania hybrydowego musi zostać potwierdzona w kolejnych badaniach.

Inną modyfikacją klasyfikatora SFk-NN/C, potencjalnie mogącą pozytywnie wpłynąć na jakość klasyfikacji, jest zastąpienie wstępnej selekcji cech współczynnikiem wagi dla cech. Dobór tego współczynnika będzie przedmiotem kolejnych badań.

Bibliografia

[1] Aeberhard S., Coomans D., De Vel O., *Comparison of classifiers in high dimensional settings*, Dept. Math. Statist., James Cook Univ., North Queensland, Australia, Tech. Rep, 92, 02, 1992.

[2] Antal B., Hajdu A., *An ensemble-based system for automatic screening of diabetic retinopathy*, Knowledge-based systems, 60, 20–27, 2014.

[3] Ayres-de Campos D., Bernardes J., Garrido A., Marques-de Sa J., Pereira-Leite L., *Sisporto 2.0: a program for automated analysis of cardiotocograms*, Journal of Maternal-Fetal Medicine, 9(5), 311–318, 2000.

[4] Bedossa P., Dargère D., Paradis V., *Sampling variability of liver fibrosis in chronic hepatitis c*, Hepatology, 38(6), 1449–1457, 2003.

[5] Berthold M.R., Cebron N., Dill F., Gabriel T.R., Kötter T., Meinl T., Ohl P., Sieb C., Thiel K., Wiswedel B., *KNIME: The Konstanz Information Miner*, Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007), Springer, 2007.

[6] Bookstein A., Kulyukin V.A., Raita T., *Generalized hamming distance*, Information Retrieval, 5(4), 353–375, 2002.

[7] Bradley A.P., *The use of the area under the roc curve in the evaluation of machine learning algorithms*, Pattern Recogn., 30(7), 1145–1159, 1997.

[8] Breiman L., Friedman J., Stone C., Olshen R., Classification and Regression Trees, The Wadsworth and Brooks-Cole statistics-probability series, Taylor & Francis, 1984.

[9] Cha S.H., *Comprehensive survey on distance/similarity measures between probability density functions*, 2007.

[10] Chapelle O., Keerthi S., Chapelle O., Keerthi S., *Multi-class feature selection with support vector machines*, 2008.

[11] Chen P.Y., Popovich P.M., Correlation: Parametric and nonparametric measures, 137-139, Sage, 2002.

[12] Czarnowski I., Jedrzejowicz P., *Ensemble classifier for mining data streams, in: 18th International Conference in Knowledge-Based and Intelligent Information and Engineering Systems, KES-2014 Gdynia, Poland, 15–17 September*, 397–406, 2014.

[13] Demšar J., *Statistical comparisons of classifiers over multiple data sets*, Journal of Machine learning research, 7(Jan), 1–30, 2006.

[14] Di Gesu V., Lo Bosco G., Pinello L., *A one class knn for signal identification: a biological case study*, International Journal of Knowledge Engineering and Soft Data Paradigms, 1(4), 376–389, 2009.

[15] Dietterich T.G., *Ensemble methods in machine learning*, Proceedings of the First International Workshop on Multiple Classifier Systems, MCS '00, 1–15, Springer-Verlag, London, UK, UK, 2000.

[16] Frank E., Witten I.H., *Generating accurate rule sets without global optimization*, 144–151, Morgan Kaufmann, 1998.

[17] Galar M., Fernandez A., Barrenechea E., Bustince H., Herrera F., *A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches*, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(4), 463–484, 2012.

[18] Geler Z., Kurbalija V., Radovanovic M., Ivanovic M., *Comparison of different weighting schemes for the knn classifier on time-series data*, Knowledge and Information Systems, 1–48, 2015.

[19] Gori M., Scarselli F., *Are multilayer perceptrons adequate for pattern recognition and verification?*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(11), 1121–1132, 1998.

[20] Gou J., Du L., Zhang Y., Xiong T., *A new distance-weighted k -nearest neighbor classifier*, 9, 1429—1436, 2011.

[21] Gramacki J., Gramacki A., *Wybrane metody redukcji wymiarowości danych oraz ich wizualizacji*, XIV Konferencja PLOUG Szczyrk, 2008.

[22] Gupta M.R., Chen Y., et al., *Theory and use of the em algorithm*, Foundations and Trends® in Signal Processing, 4(3), 223–296, 2011.

[23] Hall M., *Correlation-based feature subset selection for machine learning*, Thesis submitted in partial fulfillment of the requirements of the degree of Doctor of Philosophy at the University of Waikato, 1998.

[24] Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., Witten I.H., *The WEKA data mining software: an update*, SIGKDD Explorations, 11(1), 10–18, 2009.

[25] Harris E., *Information gain versus gain ratio: A study of split method biases.*, ISAIM, 2002.

[26] Ho T.K., Nearest neighbors in random subspaces, 640–648, Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[27] Hoeting J.A., Madigan D., Raftery A.E., Volinsky C.T., *Bayesian model averaging: a tutorial*, Statistical science, 382–401, 1999.

[28] Honaker J., King G., *What to do about missing values in time-series cross-section data*, American Journal of Political Science, 54(2), 561–581, 2010.

[29] Honaker J., King G., Blackwell M., et al., *Amelia ii: A program for missing data*, Journal of statistical software, 45(7), 1–47, 2011.

[30] Huang M., Lin R., Huang S., Xing T., *A novel approach for precipitation forecast via improved k-nearest neighbor algorithm*, Advanced Engineering Informatics, 33(Supplement C), 89–95, 2017.

[31] Jain A.K., Dubes R.C., Algorithms for clustering data, Prentice-Hall, Inc., 1988.

[32] Jóźwik A., i Inżynierii Biomedycznej im. Macieja Nałęcza P.A.N.I.B., Nieparametryczne metody klasyfikacji nadzorowanej, Zespół Wydawniczo-Poligraficzny IBIB PAN, 2013.

[33] Jurek A., Bi Y., Wu S., Nugent C., *A survey of commonly used ensemble-based classification techniques*, 29, 551–581, 2013.

[34] Khozeimeh F., Alizadehsani R., Roshanzamir M., Khosravi A., Layegh P., Nahavandi S., *An expert system for selecting wart treatment method*, Computers in biology and medicine, 81, 167–175, 2017.

[35] Kira K., Rendell L.A., *A practical approach to feature selection*, Proceedings of the ninth international workshop on Machine learning, 249–256, 1992.

[36] Kontorovich A., Weiss R., *A bayes consistent 1-nn classifier*, CoRR, abs/1407.0208, 2014, `1407.0208`, URL `http://arxiv.org/abs/1407.0208`.

[37] Koziarski M., Krawczyk B., Woźniak M., *The deterministic subspace method for constructing classifier ensembles*, Pattern Analysis and Applications, 20(4), 981–990, 2017.

[38] Krzyśko M., Wołyński W., Górecki T., Skorzybut M., Systemy uczące się, WNT, 2008.

[39] Little R.J.A., Rubin D.B., Statistical Analysis with Missing Data, John Wiley & Sons, Inc., New York, NY, USA, 1986.

[40] Maldonado S., Weber R., Basak J., *Simultaneous feature selection and classification using kernel-penalized support vector machines*, Information Sciences, 181(1), 115 – 128, 2011.

[41] Mangasarian O.L., Street W.N., Wolberg W.H., *Breast cancer diagnosis and prognosis via linear programming*, Operations Research, 43(4), 570–577, 1995.

[42] McDonald G.C., *Ridge regression*, Wiley Interdisciplinary Reviews: Computational Statistics, 1(1), 93–100, 2009.

[43] Napierała K., Stefanowski J., Wilk S., Learning from Imbalanced Data in Presence of Noisy and Borderline Examples, 158–167, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[44] Orczyk T., Porwik P., *Investigation of the impact of missing value imputation methods on the k-nn classification accuracy*, Computational Collective Intelligence, 557–565, 2015 (10 pkt. MNiSW).

[45] Orczyk T., Porwik P., Musialik J., *Ensemble of single-parameter classifiers in liver fibrosis stage recognition*, Journal of Medical Imaging and Health Informatics, 5(6), 1281–1286, 2015 (15 pkt. MNiSW, IF=0,621).

[46] Özgür A., Özgür L., Güngör T., Text Categorization with Class-Based and Corpus-Based Keyword Selection, 606–615, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[47] Porwik P., Orczyk T., Lewandowski M., Cholewa M., *Feature projection k-nn classifier model for imbalanced and incomplete medical data*, Biocybernetics and Biomedical Engineering, 36(4), 644–656, 2016 (15 pkt. MNiSW, IF=1,031).

[48] Quinlan J.R., *Induction of decision trees*, Machine Learning, 1(1), 81–106, 1986.

[49] Regev A., Berho M., Jeffers L.J., Milikowski C., Molina E.G., Pyrsopoulos N.T., Feng Z.Z., Reddy K.R., Schiff E.R., *Sampling error and intraobserver variation in liver biopsy in patients with chronic hcv infection*, The American journal of gastroenterology, 97(10), 2614, 2002.

[50] Robnik-Šikonja M., Kononenko I., *Theoretical and empirical analysis of relieff and rrelieff*, Machine Learning, 53(1), 23–69, 2003.

[51] Rubin L.H., Witkiewitz K., Andre J.S., Reilly S., *Methods for handling missing data in the behavioral neurosciences: Don't throw the baby rat out with the bath water*, J Undergrad Neurosci Educ, 5(2), A71–A77, 2007.

[52] Salunkhe U.R., Mali S.N., *Classifier ensemble design for imbalanced data classification: A hybrid approach*, Procedia Computer Science, 85, 725–732, 2016.

[53] Salzberg S.L., *C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993*, Machine Learning, 16(3), 235–240, 1994.

[54] Schafer J.L., Analysis of incomplete multivariate data, Monographs on statistics and applied probability ; 72., Chapman & Hall, London ; New York, 1997.

[55] Schafer J.L., Graham J.W., *Missing data: our view of the state of the art.*, Psychological methods, 7(2), 147, 2002.

[56] Scheffer J., *Dealing with missing data*, Research Letters in the Information and Mathematical Sciences, 153–160, 2002.

[57] Sigillito V.G., Wing S.P., Hutton L.V., Baker K.B., *Classification of radar returns from the ionosphere using neural networks*, Johns Hopkins APL Technical Digest, 10(3), 262–266, 1989.

[58] Simeone P., Marrocco C., Tortorella F., *Design of reject rules for ecoc classification systems*, Pattern Recognition, 45(2), 863–875, 2012.

[59] Skurichina M., Duin R.P.W., *Bagging, boosting and the random subspace method for linear classifiers*, Pattern Analysis & Applications, 5(2), 121–135, 2002.

[60] Srivastava M.S., Joshi M.N., Gaur M.M., *A review paper on feature selection methodologies and their applications*, International Journal of Engineering Research and Development, 7(6), 57–61, 2013.

[61] Sun Y., Kamel M.S., Wong A.K., Wang Y., *Cost-sensitive boosting for classification of imbalanced data*, Pattern Recognition, 40(12), 3358–3378, 2007.

[62] Sun Y., Wong A.K.C., Kamel M.S., *Classification of imbalanced data: A review*, International Journal of Pattern Recognition and Artificial Intelligence, 23(04), 687–719, 2009.

[63] Thai-Nghe N., Gantner Z., Schmidt-Thieme L., *Cost-sensitive learning methods for imbalanced data*, The 2010 International Joint Conference on Neural Networks (IJCNN), 1–8, 2010.

[64] Tibshirani R., *Regression shrinkage and selection via the lasso*, Journal of the Royal Statistical Society, Series B, 58, 267–288, 1994.

[65] Ting K.M., Confusion Matrix, 209–209, Springer US, Boston, MA, 2010.

[66] Van Asch V., *Macro- and micro-averaged evaluation measure*, 2013.

[67] Veni C.V.K., Rani T.S., *Ensemble based classification using small training sets : A novel approach*, 2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL), 1–8, 2014.

[68] Wang J., Xu M., Wang H., Zhang J., *Classification of imbalanced data by using the smote algorithm and locally linear embedding*, Signal Processing, 2006 8th International Conference on, tom 3, IEEE, 2006.

[69] Wang X., Matwin S., Japkowicz N., Liu X., Cost-Sensitive Boosting Algorithms for Imbalanced Multi-instance Datasets, 174–186, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[70] Weiss S.M., Indurkhya N., *Rule-based machine learning methods for functional prediction*, Journal of Artificial Intelligence Research, 3, 383–403, 1995.

[71] Witten I.H., Frank E., Hall M.A., Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd wyd., 2011.

[72] Yijing L., Haixiang G., Xiao L., Yanan L., Jinling L., *Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data*, Knowledge-Based Systems, 94, 88–104, 2016.

[73] Yousef M., Najami N., Khalifav W., *A comparison study between one-class and two-class machine learning for microrna target detection*, Journal of Biomedical Science and Engineering, 3(03), 247, 2010.

[74] Ziemba P., *Redukcja wymiarowości i selekcja cech w zadaniach klasyfikacji i regresji z wykorzystaniem uczenia maszynowego*, Studia Informatica, (30), 221–236, 2012.

Dodatek A

Wykaz oznaczeń stosowanych w rozprawie

Jeśli w tekście nie wykazano inaczej, stosowane symbole należy rozumieć jako:

X – przestrzeń cech,

$\mathbf{x}$ – obiekt z przestrzeni X,

$x_q^{(f)}$ – f-ta cecha q-tego obiektu,

$\mathbf{X}$ – zbiór obiektów $\mathbf{x}$,

F – liczba cech obiektu,

Q – liczba obiektów,

C – przestrzeń klas,

c – etykieta klasy,

c_q – etykieta klasy q-tego obiektu,

$\mathbf{R}$ – zbiór par referencyjnych $(\mathbf{x}_q, c_q)$,

L – liczba klas,

D – algorytm klasyfikacji,

$D(\mathbf{x})$ – decyzja klasyfikatora D dla obiektu $\mathbf{x}$,

$P(c|\mathbf{x})$ – prawdopodobieństwo warunkowe *a posteriori*,

$\rho(\mathbf{x}_i,\mathbf{x}_j)$ – odległość (niepodobieństwo) pomiędzy i-tym i j-tym obiektem.

Dodatek B

Spis tabel

Dodatek C

Spis rysunków